Causal Message Sequence Charts

Thomas Gazagnaire, Blaise Genest, Loïc Hélouët, P.S. Thiagarajan, Shaofa Yang

Causal Message Sequence Charts

Thomas Gazagnaire [1]

Citrix Systems R&D Ltd., UK

Blaise Genest

IRISA/CNRS, France

Loïc Hélouët

IRISA/INRIA, France

P.S. Thiagarajan

School of Computing, National University of Singapore

Shaofa Yang [1]

NUS, Singapore

Abstract

Scenario languages based on Message Sequence Charts (MSCs) have been widely studied in the last decade [21,20,3,15,12,19,14]. The high expressive power of MSCs renders many basic problems concerning these languages undecidable. However, several of these problems are decidable for languages that possess a behavioral property called "existentially bounded". Unfortunately , collections of scenarios outside this class are frequently exhibited by systems such as sliding window protocols. We propose here an extension of MSCs called causal Message Sequence Charts and a natural mechanism for defining languages of causal MSCs called causal HMSCs (CaHMSCs). These languages preserve decidable properties without requiring existential bounds. Further, they can model collections of scenarios generated by sliding window protocols. We establish here the basic theory of CaHMSCs as well as the expressive power and complexity of decision procedures for various subclasses of CaHMSCs. We also illustrate the modeling power of our formalism with the help of a realistic example based on the TCP sliding window feature.

Preprint submitted to Elsevier *2 December 2008*

1 Introduction

Scenario languages based on Message Sequence Charts (MSCs) have met considerable interest in the last decade. The attractiveness of this notation derives from two of its major characteristics. Firstly, MSCs have a simple and appealing graphical representation based on just a few concepts: processes, messages and internal actions. Secondly, from a mathematical standpoint, scenario languages admit an elegant formalization: they can be defined as languages generated by finite-state automata over an alphabet of MSCs. These automata are usually called High-level Message Sequence Charts (HMSCs) [16].

An MSC is a restricted kind of labelled partial order and an HMSC is a generator of a set of MSCs, that is, a language of MSCs. For example, the MSC M shown in Figure 2 is a member of the MSC language generated by the HMSC of Figure 1 while the MSC N shown in Figure 2 is not.

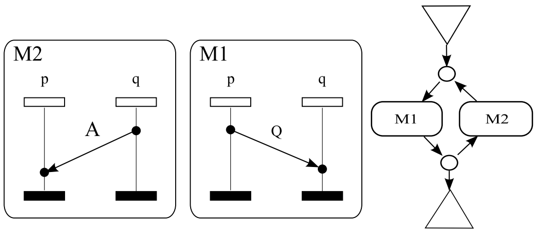

Fig. 1. An HMSC over two MSCs

HMSCs are very expressive and hence a number of basic problems associated with them cannot be solved effectively. For instance, it is undecidable whether two HMSCs generate the same collection of MSCs [21] or whether an HMSC generates a regular MSC language; an MSC language is regular if the collection of all the linearizations of all the MSCs in the language is a regular string language in the usual sense. Consequently, subclasses of HMSCs have been identified [20,3,12] and studied.

On the other hand, a basic limitation of HMSCs is that their MSC languages are finitely generated. More precisely, each MSC in the language can be defined as the sequential composition of elements chosen from a fixed finite set of MSCs [19]. However, the behaviors of many protocols constitute MSC languages that are *not* finitely generated. This occurs for example with scenarios generated by the alternating bit protocol. Such protocols can induce a collection of braids like N in Figure 2 which cannot be finitely generated.

Email addresses: thomas.gazagnaire@citrix.com (Thomas Gazagnaire), blaise.genest@irisa.fr (Blaise Genest), loic.helouet@irisa.fr (Loïc Hélouët), thiagu@comp.nus.edu.sg (P.S. Thiagarajan), yangsf@comp.nus.edu.sg (Shaofa Yang).

[1] Work done while this author was at IRISA/INRIA, France.

One way to handle this is to work with the so called safe *compositional* HMSCs in which message emissions and receptions are decoupled in individual MSCs but matched up at the time of composition, so as to yield an MSC. Compositional HMSCs are however notationally awkward and do not possess the visual appeal of HMSCs. Furthermore the general class of compositional HMSC languages embeds the full expressive power of communicating automata [5] and consequently inherits all their undecidability results.

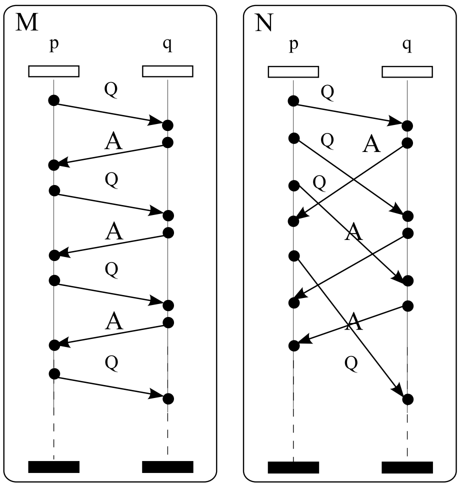

Fig. 2. Two MSCs M and N

This paper proposes a new approach to increase the modeling power of HMSCs in a tractable manner. We first extend the notion of an MSC to a *causal* MSC in which the events belonging to each lifeline (process), instead of being linearly ordered, are allowed to be partially ordered. To gain modelling power, we do not impose any serious restrictions on the nature of this partial order. However, we assume a suitable Mazurkiewicz trace alphabet [8] for each lifeline and use this to define a composition operation for causal MSCs. This leads to the notion of causal HMSCs.

A property called *existential boundedness* [11] leads to a powerful proof technique for establishing decidability results for HMSCs. Informally, this property states that there is a uniform upper bound K such that for every MSC in the language *there exists* an execution along which -from start to finish- all FIFO channels remain K-bounded. On the other hand, when *all executions* of all MSCs in a language can occur within a fixed upper bound K on channels, the corresponding property is called *universally bounded*. A causal HMSC (i.e. the MSC language associated with a causal HMSC) is *a priori* not existentially bounded. Hence the proof method cited above can not be used to obtain the desired decidablity results. Instead, we need to generalize the methods of [20] and of [12] in a non-trivial way.

Our fist major result is to formulate natural -and decidable- structural conditions and to show that causal HMSCs satisfying these conditions generate

MSC languages that are regular. Our second major result is that the inclusion problem for causal HMSCs (i.e. given two causal HMSCs, whether the MSC language defined by the first one is included in the MSC langauge of the other) is decidable for causal HMSCs using the same Mazurkiewicz trace alphabets, provided at least one of them has the structural property known as *globally-cooperative*. Furthermore, we prove that the restriction that the two causal HMSCs have identical Mazurkiewicz trace alphabets associated with them is necessary. These results constitute a non-trivial extension for causal HMSCs of comparable results on HMSCs [20,3,14] and [12]. In addition, we identify the property called "window-bounded" which appears to be an important ingredient of the "braid"-like MSC languages generated by many protocols. Basically, this property bounds the number of messages a process p can send to a process q before having received an acknowledgement to the earliest message. We show it is decidable if a causal HMSC generates a window-bounded MSC language. Finally, we compare the expressive power of languages based on causal HMSCs with other known HMSC-based language and give a detailed example based on the TCP protocol to illustrate the modeling potential of causal HMSCs.

This paper is an extended version of the work presented in [10] and contains several important changes and improvements. Specifically, the definition of s-regularity for causal HMSCs has been weakened. As a result, causal HMSCs which were not s-regular according to [10] -and in fact not even globally-coperative- are deemed to be s-regular under the weakened definition. We have also included here complete proofs and have added a detailed example.

In the next section we introduce causal MSCs and causal HMSCs. We also define the means for associating an ordinary MSC language with a causal HMSC. In the subsequent section we develop the basic theory of causal HMSCs. To this end, we identify the subclasses of s-regular (syntactically regular) and globally-cooperative causal HMSCs and develop our decidability results. In section 4, we identify the "window-bounded" property, and show that one can decide if a causal HMSC generates a window-bounded MSC language. In section 5 we compare the expressive power of languages based on causal HMSCs with other known HMSC-based language classes. Finally, in section 6, we give a detailed example based on the TCP protocol to illustrate the modeling potential of causal HMSCs.

2 MSCs, causal MSCs and causal HMSCs

Through the rest of the paper, we fix a finite nonempty set $\mathcal{P}$ of process names with $|\mathcal{P}| > 1$. For convenience, we let p, q range over $\mathcal{P}$ and drop the subscript $p \in \mathcal{P}$ when there is no confusion. We also fix finite nonempty sets *Msg*, *Act* of message types and internal action names respectively. We define the alphabets $\Sigma_! = \{p!q(m) \mid p, q \in \mathcal{P}, p \neq q, m \in Msg\}$, $\Sigma_? = \{p?q(m) \mid p, q \in \mathcal{P}, p \neq q, m \in Msg\}$, and $\Sigma_{act} = \{p(a) \mid p \in \mathcal{P}, a \in Act\}$. The letter $p!q(m)$ means the sending of a message with content m from p to q; $p?q(m)$ the reception of a message of content m at p from q; and $p(a)$ the execution of an internal action a by process p. Let $\Sigma = \Sigma_! \cup \Sigma_? \cup \Sigma_{act}$. We define the *location* of a letter α in Σ, denoted $loc(\alpha)$, by $loc(p!q(m)) = p = loc(p?q(m)) = loc(p(a))$. For each process p in $\mathcal{P}$, we set $\Sigma_p = \{\alpha \in \Sigma \mid loc(\alpha) = p\}$.

Definition 1 *A* causal MSC *over* $(\mathcal{P}, \Sigma)$ *is a structure* $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$, *where* E *is a finite nonempty set of* events, $\lambda : E \to \Sigma$ *is a labelling function, and the following conditions hold:*

- *For each process* p, $\sqsubseteq_p \subseteq E_p \times E_p$ *is a* partial order, *where* $E_p = \{e \in E \mid \lambda(e) \in \Sigma_p\}$. *We let* $\widehat{\sqsubseteq}_p \subseteq E_p \times E_p$ *denote the least relation such that* $\sqsubseteq_p$ *is the reflexive and transitive closure of* $\widehat{\sqsubseteq}_p$ *(*$\widehat{\sqsubseteq}_p$ *is the Hasse diagram of* $\sqsubseteq_p$*).*
- $\ll \subseteq E_! \times E_?$ *is a bijection, where* $E_! = \{e \in E \mid \lambda(e) \in \Sigma_!\}$ *and* $E_? = \{e \in E \mid \lambda(e) \in \Sigma_?\}$. *For each* $e \ll e'$, $\lambda(e) = p!q(m)$ *iff* $\lambda(e') = q?p(m)$.
- *The transitive closure* $\leq$ *of the relation* $\left(\bigcup_{p \in \mathcal{P}} \sqsubseteq_p\right) \cup \ll$ *is a partial order.*

For each p, the relation $\sqsubseteq_p$ dictates the "causal" order in which events of E_p may be executed. The relation $\ll$ identifies pairs of message-emission and message-reception events. We say that the causal MSC B is *weak-FIFO* iff for any $e \ll f$, $e' \ll f'$ such that $\lambda(e) = \lambda(e') = p!q(m')$ (and thus $\lambda(f) = \lambda(f') = q?p(m)$), we have either $e \sqsubseteq_p e'$ and $f \sqsubseteq_q f'$; or $e' \sqsubseteq_p e$ and $f' \sqsubseteq_q f$. In weak-FIFO [2] scenarios, messages of the same content between two given processes cannot overtake. Note however that messages of different kind between two processes can overtake. Note that we do not demand *a priori* that a causal MSC must be weak FIFO. Testing (weak) fifoness of a causal MSC of size b can be done in at most $\mathcal{O}(\frac{b^2}{8} - \frac{b}{4})$, by considering all pairs of messages in the MSC.

Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ be a causal MSC. We shall write $|B|$ for $|E|$ and refer to $|B|$ as the *size* of B. A *linearization* of B is a word $a_1 a_2 \ldots a_\ell$ over Σ such that $E = \{e_1, \ldots, e_\ell\}$ with $\lambda(e_i) = a_i$ for each i; and $e_i \leq e_j$ implies $i \leq j$ for any i, j. We let $Lin(B)$ denote the set of linearizations of B. Clearly, $Lin(B)$ is nonempty. We set $Alph(B) = \{\lambda(e) \mid e \in E\}$, and $Alph_p(B) = Alph(B) \cap \Sigma_p$ for each p.

The leftmost part of Figure 3 depicts a causal MSC M. In this diagram, we enclose events of each process p in a vertical box and show the partial order $\sqsubseteq_p$ in the standard way. In case $\sqsubseteq_p$ is a total order, we place events of p along a verti-

[2] A different notion called strong FIFO that does not allow overtakings of messages between two given processes of different content is also frequently used in the MSC literature.

cal line with the minimum events at the top and omit the box. In particular, in M, the two events on p are not ordered (i.e. $\widehat{\sqsubseteq}_p$ is empty) and $\sqsubseteq_q$ is a total order. Members of $\ll$ are indicated by horizontal or downward-sloping arrows labelled with the transmitted message. Both words $p!q(Q).q!p(A).q?p(Q).p?q(A)$ and $q!p(A).p?q(A).p!q(Q).q?p(Q)$ are linearizations of M.

An *MSC* $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ is defined in the same way as a causal MSC except that every $\sqsubseteq_p$ is required to be a *total order*. In an MSC B, the relation $\sqsubseteq_p$ must be interpreted as the visually observed order of events in one sequential execution of p. Let $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ be a causal MSC. Then we say the MSC B is a *visual extension* of B' if $E' = E$, $\lambda' = \lambda$, $\sqsubseteq'_p \subseteq \sqsubseteq_p$ and $\ll' = \ll$. We let $Vis(B')$ denote the set of visual extensions of B'.

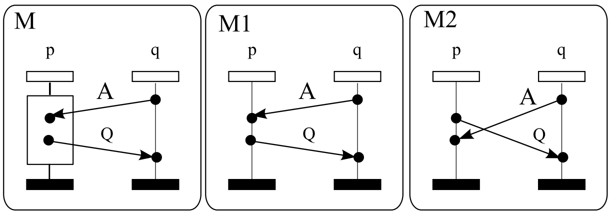

Fig. 3. A causal MSC M and its visual extensions $M1, M2$.

In Figure 3, $Vis(M)$ consists of MSCs $M1, M2$. The initial idea of visual ordering comes from [2], that notices that depending on the interpretation of an MSC, for example when a lifeline describes a physical entity in a network, imposing an ordering on message receptions is not possible. Hence, [2] distinguishes two orderings on MSCs: a visual order (that is the usual order used for MSCs), that comes from the relative order of events along an instance line, and a causal order, that is weaker, and does not impose any ordering among consecutive receptions.

Note that the set of visual extensions of a causal MSC B is not necessarily the union of instance per instance linearizations, as an extension of a causal MSC must remain a MSC, ie. the relation among the events has to remain a partial order. Consider for example, the causal MSC B of Figure 4, and its visual extensions in Figure 5: in any visual extension $V = (\{e_1, e_2, f_1, f_2\}, \lambda, \{\sqsubseteq_p, \sqsubseteq_q\}, \ll) \in Vis(B)$ we cannot have $e_2 \sqsubseteq_p e_1$ and $f_1 \sqsubseteq_q f_2$ at the same time.

We next define a concatenation operation for causal MSCs. Unlike for usual MSCs, events of a same process need not be dependent. To express whether there should be a dependency or not, for each process p in $\mathcal{P}$, we fix a concurrent alphabet (Mazurkiewicz trace alphabet [8]) (Σ_p, I_p) for each process $p \in \mathcal{P}$, where $I_p \subseteq \Sigma_p \times \Sigma_p$ is a symmetric and irreflexive relation called the *independence relation* over the alphabet of actions Σ_p. We denote the *dependence relation* $(\Sigma_p \times \Sigma_p) - I_p$ by D_p. These relations are fixed *for the rest of the paper* (unless explicitly stated otherwise). Following the usual definitions of Mazurkiewicz traces, for each (Σ_p, I_p), the associated trace equivalence re-

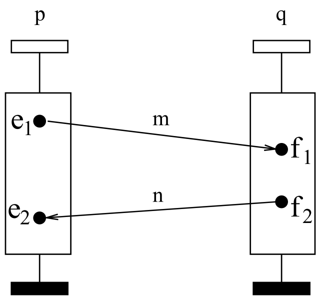

Fig. 4. An example of causal MSC B: the set of visual extensions of B is not the instance per instance commutative closure of any visual extension of B.

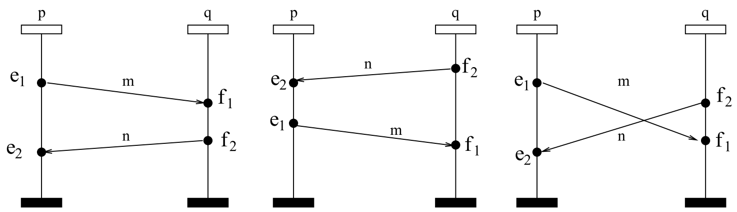

Fig. 5. Visual extensions of the causal MSC B of Figure 4.

lation $\sim_p$ over $\Sigma_p^\star$ is the least equivalence relation such that, for any u, v in $\Sigma_p^\star$ and α, β in Σ_p, $\alpha\ I_p\ \beta$ implies $u\alpha\beta v \sim_p u\beta\alpha v$. Equivalence classes of $\sim_p$ are called *traces*. For u in $\Sigma_p^\star$, we let $[u]_p$ denote the trace containing u.

Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ be a causal MSC. We say $\sqsubseteq_p$ *respects* the trace alphabet (Σ_p, I_p) iff for any $e, e' \in E_p$, the following hold:

(i) $\lambda(e)\ D_p\ \lambda(e')$ implies $e\ \sqsubseteq_p e'$ or $e'\ \sqsubseteq_p e$
(ii) $e\ \widehat{\sqsubseteq}_p\ e'$ implies $\lambda(e)\ D_p\ \lambda(e')$

A causal MSC B is said to respect the trace alphabets $\{(\Sigma_p, I_p)\}_{p \in \mathcal{P}}$ iff $\sqsubseteq_p$ respects (Σ_p, I_p) for every p. In order to gain modelling power, we have allowed each $\sqsubseteq_p$ to be *any* partial order, not necessarily respecting (Σ_p, I_p).

We shall now define the concatenation operation of causal MSCs using the trace alphabets $\{(\Sigma_p, I_p)\}_{p \in \mathcal{P}}$.

Definition 2 *Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ and $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ be causal MSCs. We define the* concatenation *of B with B', denoted by $B \circledcirc B'$, as the causal MSC $B'' = (E'', \lambda'', \{\sqsubseteq''_p\}, \ll'')$ where:*

- *E'' is the disjoint union of E and E'. λ'' is given by: $\lambda''(e) = \lambda(e)$ if $e \in E$, $\lambda''(e) = \lambda'(e)$ if $e \in E$ and $\ll'' = \ll \cup \ll'$.*
- *For each p, $\sqsubseteq''_p$ is the transitive closure of*

$$\sqsubseteq_p \bigcup \sqsubseteq'_p \bigcup \{(e, e') \in E_p \times E'_p \mid \lambda(e)\ D_p\ \lambda'(e')\}$$

Clearly, $\circledcirc$ is a well-defined and associative operation. Note that in case B and B' are MSCs and $D_p = \Sigma_p \times \Sigma_p$ for every p, then the result of $B \circledcirc B'$ is the asynchronous concatenation (also called weak sequential composition) of B with B' [22], which we denote by $B \circ B'$. Note that when B_1 and B_2 are weak FIFO causal MSCs, then their concatenation is also weak FIFO. This property comes from the irreflexive nature of the independence relations. This remark also holds for the concatenation of MSCs. We also remark that the concatenation of causal MSCs is different from the concatenation of traces. The concatenation of trace $[u]_p$ with $[v]_p$ is the trace $[uv]_p$. However, a causal MSC B needs not respect $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$. Consequently, for a process p, the projection of $Lin(B)$ on $Alph_p(B)$ is *not* necessarily a trace.

Figure 6 shows an example of sequential composition of two causal MSCs B_1 and B_2, with the dependency relations D_p and D_q being the commutative and reflexive closure of $\Big\{\big(p!q(m), p!q(n)\big), \big(p!q(n), p?q(u)\big)\Big\}$ and $\Big\{\big(q?p(n), q!p(v)\big)\Big\}$ respectively. Note that although the dependence relation D_p contains the pair $\big(p!q(m), p!q(n)\big)$, sendings of messages m and n by process p in B_1 can be unordered, and remain unordered after composition.

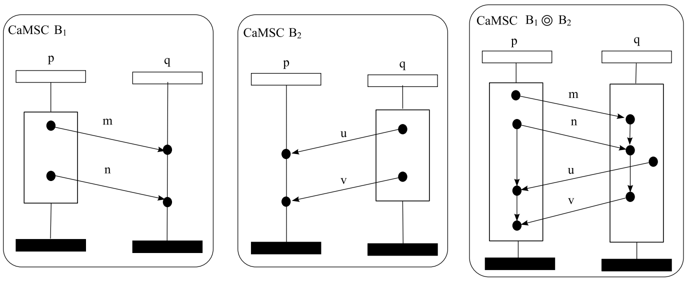

Fig. 6. Concatenation example.

An usual way to extend the composition mechanism is to use an automaton labelled by basic scenarios, to produce scenario languages. These automata are called High-level Message Sequence Charts (or HMSCs for short)[25,23] when MSCs are concatenated, and a similar construct exists for compositional Message Sequence Charts [13,11,7]. We can now define causal HMSCs. Recall that we have fixed a set $\mathcal{P}$ of process names and a family $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$ of Mazurkiewicz trace alphabets.

Definition 3 *A* causal HMSC *over* $(\mathcal{P}, \{(\Sigma_p, I_p)\}_{p\in\mathcal{P}})$ *is a structure* $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ *where N is a finite nonempty set of* nodes*, $N_{in} \subseteq N$ the set of initial nodes, $\mathcal{B}$ a finite nonempty set of causal MSCs, $\longrightarrow \subseteq N \times \mathcal{B} \times N$ the transition relation, and $N_{fi} \subseteq N$ the set of final nodes.*

A *path* in the causal HMSC H is a sequence $\rho = n_0 \xrightarrow{B_1} n_1 \xrightarrow{B_2} \cdots \xrightarrow{B_\ell} n_\ell$. If $n_0 = n_\ell$, then we say ρ is a cycle. The path ρ is *accepting* iff $n_0 \in N_{in}$ and $n_\ell \in N_{fi}$. The causal MSC generated by ρ, denoted $\circledcirc(\rho)$, is $B_1 \circledcirc B_2 \circledcirc \cdots \circledcirc B_\ell$. We let $cMSC(H)$ denote the set of causal MSCs generated by accepting paths of H. We also set $Vis(H) = \bigcup\{Vis(M) \mid M \in cMSC(H)\}$ and $Lin(H) = \bigcup\{Lin(M) \mid M \in cMSC(H)\}$. Obviously, $Lin(H)$ is also equal to $\bigcup\{Lin(M) \mid M \in Vis(H)\}$. We shall refer to $cMSC(H)$, $Vis(H)$, $Lin(H)$, respectively, as the causal language, visual language and linearization language of H.

An *HMSC* $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ is defined in the same way as a causal HMSC except that $\mathcal{B}$ is a finite set of MSCs, and that the concatenation operation used to produce MSC languages is the weak sequential composition $\circ$. HMSCs can then be considered as causal HMSCs labelled with MSCs, and equipped with empty independence relation (for every $p \in \mathcal{P}, I_p = \emptyset$). Hence, a path ρ of H generates an MSC by concatenating the MSCs along ρ with operation $\circ$. We let $Vis(H)$ denote the set of MSCs generated by accepting paths of H with $\circ$, and call $Vis(H)$ the visual language of H. Recall that an MSC language (i.e. a collection of MSCs) L is *finitely generated* [19] iff there exists a finite set X of MSCs satisfying the condition: for each MSC B in L, there exist $B_1, \ldots, B_\ell$ in X such that $B = B_1 \circ \cdots \circ B_\ell$. Many protocols exhibit scenario collections that are *not* finitely generated. For example, sliding window protocols can generate arbitrarily large MSCs repeating the communication behavior shown in MSC N of Figure 2. One basic limitation of HMSCs is that their visual languages are *finitely generated*. In contrast, the visual language of a causal HMSC is *not* necessarily finitely generated. Consider for instance the HMSC H in Figure 1, and the independence relations given by: $I_p = \{((p!q(Q), p?q(A)), (p?q(A), p!q(Q)))\}$ and $I_q = \emptyset$. $M1$ and $M2$ can be seen as causal MSCs, and H as a causal HMSC over $(\mathcal{P} = \{p, q\}, \{(\Sigma_p, I_p)(\Sigma_q, I_q)\})$. Clearly, $Vis(H)$ is not finitely generated, as it contains infinitely many MSCs similar to N of Figure 2. Throughout the paper, we will use the following standardized graphical convention to depict (causal) HMSCs: nodes are represented by circles, initial nodes are nodes connected to a downward pointing triangle, final nodes are nodes connected to an upward pointing triangle, and a transition $t = (n, B, n')$ is represented by an arrow decorated by a rectangle containing the (causal) MSC B.

3 Regularity and Language Inclusion for causal HMSCs

3.1 Semantics for causal HMSCs

As things stand, a causal HMSC H defines three syntactically different languages, namely its linearization language $Lin(H)$, its visual (MSC) language $Vis(H)$ and its causal MSC language $cMSC(H)$. The next proposition shows that they are also semantically different in general. It also identifies the restrictions under which they match semantically.

Proposition 1 *Let H, H' be two causal HMSCs over the* same *family of trace alphabets $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$. Consider the following six hypotheses:*

(i)	$cMSC(H) = cMSC(H')$	*(i)'*	$cMSC(H) \cap cMSC(H') \neq \emptyset$
(ii)	$Vis(H) = Vis(H')$	*(ii)'*	$Vis(H) \cap Vis(H') \neq \emptyset$
(iii)	$Lin(H) = Lin(H')$	*(iii)'*	$Lin(H) \cap Lin(H') \neq \emptyset$

Then we have:

- $(i) \Rightarrow (ii)$, $(i)' \Rightarrow (ii)'$, $(ii) \Rightarrow (iii)$ *and* $(ii)' \Rightarrow (iii)'$ *but the converses do not hold in general.*
- *If every causal MSC labelling transitions of H and H' respects $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$, then $(i) \Leftrightarrow (ii)$ and $(i)' \Leftrightarrow (ii)'$.*
- *If every causal MSC labelling transitions of H and H' is weak FIFO, then $(ii) \Leftrightarrow (iii)$ and $(ii)' \Leftrightarrow (iii)'$.*

Proof:

- The implications $(i) \implies (ii)$ and $(ii) \implies (iii)$ follow from the definitions. However, as shown in Figure 7, $Vis(G_1) = Vis(H_1)$ but $cMSC(G_1) \neq cMSC(H_1)$. And $Lin(G_2) = Lin(H_2)$ but $Vis(G_2) \neq Vis(H_2)$. Note that the independence relations are immaterial in these examples.
- If every causal MSC labelling transitions of H and H' respects $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$, then one can define an equivalence relation $M \equiv M'$ on MSCs iff there exists a causal MSC C with $M, M' \in Vis(C)$. Then, for any causal MSC B in $cMSC(H) \bigcup cMSC(H')$, $Vis(B)$ is an equivalence class of that relation, and $(i) \Leftrightarrow (ii)$ and $(i)' \Leftrightarrow (ii)'$.
- If every causal MSC labelling transitions of H and H' is weak FIFO, as remarked earlier, we know that all MSCs in $Vis(H) \cup Vis(H')$ are weak FIFO since the independence relations are irreflexive. Now, for each linearization w, one can reconstruct a unique weak FIFO MSC. Hence, if $Lin(M_1) = Lin(M_2)$ for M_1 and M_2 are in $Vis(H) \cup Vis(H')$, they are weak FIFO, and we necessarily have $M_1 = M_2$, and $(ii) \Leftrightarrow (iii)$ and $(ii)' \Leftrightarrow (iii)'$.

□

For most purposes, the relevant semantics for a causal HMSC seems to be its visual language. However in the following we focus first in section 3.2 on the linearization language properties of causal HMSCs. Then, we focus in section 3.3 on the causal language properties. Using the above proposition 1, it is then straightforward to translate these properties to the visual language of causal HMSCs, when the right hypothesis apply.

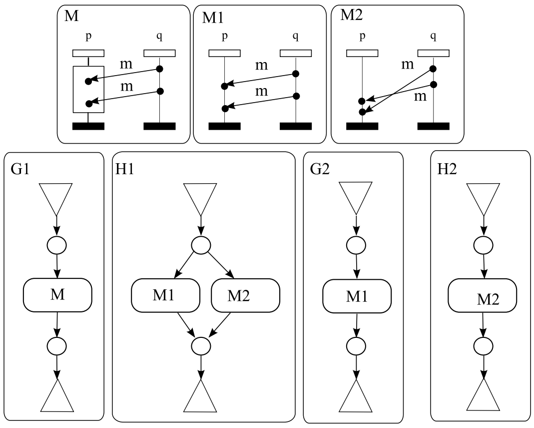

Fig. 7. Relations between linearizations, visual extensions and causal orders

3.2 Regular sets of linearizations

It is undecidable in general whether an HMSC has a regular linearization language [20]. In the literature, a subclass of HMSCs called regular [20] (or bounded [3]) HMSCs, has been identified. In this paper, to avoid overloading "regular", we shall refer to this syntactic property as "s-regular". The importance of this property lies in the fact that linearization language of every s-regular HMSC is regular. Furthermore, one can effectively decide whether an HMSC is s-regular. Our goal is to extend these results to causal HMSCs.

The key notion of characterizing s-regular HMSCs is that of the communication graph of an MSC. The communication graph captures intuitively the structure of information exchanges among processes in an MSC. Given an MSC M, its communication graph is a directed graph that has processes of M as vertices, and contains an edge from p to q if p sends a message to q somewhere in M. Given an HMSC H, we shall say that M is a cycle-MSC of H if there is a cycle in H such that M is obtained by concatenating the MSCs encountered along this cycle. H is said to be s-regular iff the communication graph of every cycle-MSC of H is strongly connected. H is said to be globally-cooperative [19] in case the communication graph of every cycle MSC of H is connected.

We can define a similar notion for causal MSCs. As processes do not necessarily impose an ordering on events, it is natural to focus on the associated Mazurkiewicz alphabet. Thus the communication graph is defined w.r.t the dependency relations $\{D_p\}_{p \in \mathcal{P}}$ used for the concatenation while the dependencies among letters of the same process are disregarded.

Definition 4 *Let* $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ *be a causal MSC. The* communication graph *of* B *is denoted by* CG_B*, and is the directed graph* $(Q, \rightsquigarrow)$*, where* $Q = \lambda(E)$ *and* $\rightsquigarrow \subseteq Q \times Q$ *is given by:* $(x, y) \in \rightsquigarrow$ *iff*

- $x = p!q(m)$ *and* $y = q?p(m)$ *for some* $p, q \in \mathcal{P}$ *and* $m \in Msg$*, or*
- xD_py.

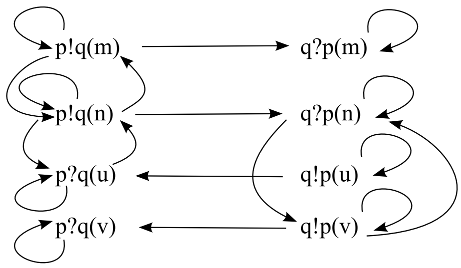

Fig. 8. Communication graph for causal MSC $B_1 \circledcirc B_2$ of Figure 6.

The example of figure 8 shows the communication graph for the causal MSC $B_1 \circledcirc B_2$ in Figure 6. For instance, there are arrows between $q?p(n)$ and $q!p(v)$ since $q?p(n)\ D_q\ q!p(v)$. However, there is no arrow between $q?p(m)$ and $q?p(n)$, even though some events of $B_1 \circledcirc B_2$ labeled by $q?p(m)$ and $q?p(n)$ are dependent. Note that for a pair of causal MSCs B, B', the communication graph $CG_{B \circledcirc B'} = (Q, \rightsquigarrow)$ can be computed from the communication graphs $CG_B = (Q_B, \rightsquigarrow_B)$ and $CG_{B'} = (Q_{B'}, \rightsquigarrow_{B'})$ as follows: $Q = Q_B \cup Q_{B'}$ and $\rightsquigarrow = \rightsquigarrow_B \cup \rightsquigarrow_{B'} \cup \Big(Q^2 \cap \bigcup\limits_{p \in \mathcal{P}} D_p\Big)$. Hence, for a fixed set of independence relations, if a causal MSC B is obtained by sequential composition, that is $B = B_1 \circledcirc B_2 \circledcirc \cdots \circledcirc B_k$, then the communication graph of B does not depend on the respective ordering of $B_1, \cdots B_k$, nor on the number of occurrences of each B_i. Hence, for any permutation f on $1..k$ and any $B' = B_{f(1)} \circledcirc B_{f(2)} \circledcirc \cdots \circledcirc B_{f(k)}$, we have that $CG_B = CG_{B'}$.

In the sequel, we will say that the causal MSC B is *tight* iff its communication graph CG_B is weakly connected. We say that B is *rigid* iff its communication graph is strongly connected. We will focus here on rigidity and study the notion of tightness in section 3.3.

Definition 5 *Let* $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ *be a causal HMSC. We say that* H *is* s-regular *(resp.* globally-cooperative*) iff for every cycle* ρ *in* H*, the causal MSC* $\circledcirc(\rho)$ *is* rigid *(resp.* tight*).*

It is easy to see that the simple protocol modeled by the causal HMSC of Figure 9 is regular, since the only elementary cycle is labeled by two local events a, b, one message from p to q and one message from q to p. The communication graph associated to this elementary cycle is strongly connected. Note that the visual language of this causal HMSC is not finitely generated, as messages m and n can cross between two occurrences of a and b.

There can be infinitely many cycles in H, hence Definition 5 does not give automatically an algorithm to check whether a causal HMSC is s-regular or globally-cooperative. However, we can use the following equivalent definition

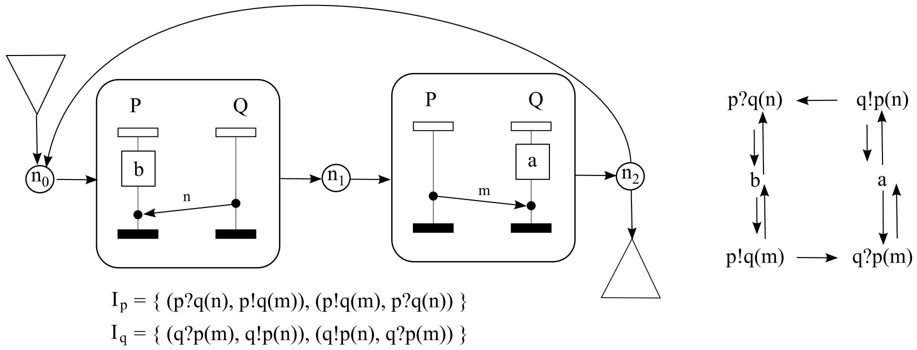

Fig. 9. A non finitely generated s-regular causal HMSC and its communication graph.

to obtain an algorithm: H is s-regular (resp. globally-cooperative) iff for every strongly connected subgraph G of H with $\{B_1, \ldots, B_\ell\}$ being the set of causal MSCs appearing in G, we have $B_1 \circledcirc \ldots \circledcirc B_\ell$ is rigid (resp. tight). As already discussed, the rigidity of $B_1 \circledcirc \ldots \circledcirc B_\ell$ does not depend on the order in which $B_1, \ldots, B_\ell$ are listed. This leads to a co-NP-complete algorithm to test whether a causal HMSC is s-regular.

Theorem 1 *Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a causal HMSC. Testing whether H is s-regular (respectively globally-cooperative) can be done in time $\mathcal{O}((|N|^2 + |\Sigma|^2) \cdot 2^{|\mathcal{B}|})$. Furthermore these problems are co-NP complete.*

Proof: We use the ideas in the proofs of [20,12], improving the deterministic complexity implied by the proof in [12], which was exponential in the number of transitions of the HMSC (there is at least one transition per label (else we can delete the useless labels)).

We first guess a subset $X = \{B_1, \cdots B_k\} \subseteq \mathcal{B}$ of causal MSCs and check that the communication graph of $B_1 \circledcirc \cdots \circledcirc B_k$ is not strongly connected (respectively disconnected). Using Tarjan's algorithm [26], this can be done in time linear in the number of edges of the communicating graph, that is quadratic in $|\Sigma|$. Then, we decompose the graph H_X into maximal strongly connected components in time $\mathcal{O}(|N|^2)$ using Tarjan again, where H_X is the restriction of H to transitions labeled by causal MSCs in X. Then it suffices to check in time $|N|^2$ whether one of this maximal strongly connected component uses all the labels from X. If it is the case, then we have a witness that H is not s-regular (resp. globally-cooperative). We thus obtain a co-NP algorithm. As there are $2^{|\mathcal{B}|}$ subsets X, this gives the deterministic time complexity.

The hardness part follows directly from the co-NP hardness result in [20]. As already mentioned, any HMSC can be seen as a causal HMSC where the independence relation of each process is empty. That is, checking whether such causal HMSC are globally-cooperative or s-regular is equivalent to checking global cooperativeness or regularity of the HMSC. These problems being co-

NP complete, we get the co-NP hardness. $\square$

Theorem 2 *Let $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ be a s-regular causal HMSC. Then $Lin(H)$ is a regular subset of $\Sigma^\star$, i.e. we can build an automaton $\mathcal{A}_H$ over Σ that recognizes $Lin(H)$. Furthermore, the number of states of $\mathcal{A}_H$ is at most in*

$$\left(|N|^2 \cdot 2^{|\Sigma|} \cdot (\Sigma + 1)^{K \cdot M} \cdot 2^{f(K \cdot M)}\right)^K ,$$

where $K = |N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|}$, $M = \max\{|B| \mid B \in \mathcal{B}\}$ (recall that $|B|$ denotes the size of the causal MSC B)nd the function f is given by $f(n) = \frac{1}{4}n^2 + \frac{3}{2}n + \mathcal{O}(\log_2 n)$.

In [18], the regularity of linearization languages of s-regular HMSC was proved by using an encoding into connected traces and building a finite state automaton which recognizes such connected traces. In our case, finding such embedding into Mazurkiewicz traces seems impossible due to the fact that causal MSCs need not be FIFO. Instead, we shall use techniques from the proof of regularity of trace closures of loop-connected automata from [8,20].

The rest of this subsection is devoted to the proof of Theorem 2. We fix a s-regular causal HMSC H as in the theorem, and show the construction of the finite state automaton $\mathcal{A}_H$ over Σ which accepts $Lin(H)$.

First, we establish some technical results.

Lemma 1 *Let $\rho = \theta_1 \ldots \theta_2 \ldots \theta_{|\Sigma|}$ be a path of H, where for each $i = 1 \ldots |\Sigma|$, the subpath $\theta_i = n_{i,0} \xrightarrow{B_{i,1}} n_{i,1} \ldots n_{i,\ell_i - 1} \xrightarrow{B_{i,\ell_i}} n_{i,0}$ is a cycle (these cycles need not be contiguous). Suppose further that the sets $\widehat{\mathcal{B}}_i = \{B_{i,1}, \ldots, B_{i,\ell_i}\}$, $i = 1, \ldots, |\Sigma|$, are equal. Let e be an event in $\circledcirc(\theta_1)$ and e' an event in $\circledcirc(\theta_{|\Sigma|})$. Let $\circledcirc(\rho) = (E, \lambda, \{\sqsubseteq_p\}, \ll)$. Then we have $e \leq e'$.*

Proof:

First of all notice that when (σ, σ') is an edge of CG_B, then for every causal MSC B', in the causal MSC $B \circledcirc B'$, every event e of B such that $\lambda(e) = \sigma$ precedes all events e' of B' such that $\lambda(e') = \sigma'$. Indeed, if σ and σ' belong to the same Σ_p, then (σ, σ') is an edge of CG_B if and only if $\sigma D_p \sigma'$, and we necessarily have $e \leq e'$ in $B \circledcirc B'$. Similarly, if σ and σ' label events located on different processes, then σ is of the form $p!q(m)$ and σ' of the form $q?p(m)$. Hence, there exists an event e'' of B such that $e \ll e''$ and $\lambda(e'') = \sigma'$. As the dependence relations are reflexive, we also have $e'' \leq e'$. Similarly, for two cycles θ_i,θ_{i+1} of ρ, if (σ, σ') is an edge of $CG_{B_{i,1} \circledcirc \cdots \circledcirc B_{i,l_i}}$, then all events in $\circledcirc(\theta_i)$ labelled by σ precede events of $\circledcirc(\theta_{i+1})$ labelled by σ'. As H is rigid, $CG_{B_{i,1} \circledcirc \cdots B_{i,l_i}}$ is strongly connected, and contains a path $(\sigma_1, \sigma_2) \cdots (\sigma_{k-1}, \sigma_k)$ of length at most $|\Sigma|$ from $\sigma_1 = \lambda(e)$ to $\sigma_k = \lambda(e')$, and we can find one event e_i,

$i \in 1..|\Sigma|$ for each cycle such that $\lambda(e_i) = \sigma_i$ and $e = e_1 \leq e_2 \leq \cdots \leq e_{|\Sigma|} = e'$. $\square$

Let $\rho = n_0 \xrightarrow{B_1} \cdots \xrightarrow{B_\ell} n_\ell$ be a path in H, where $B_i = (E_i, \lambda_i, \{\sqsubseteq_p^i\}, \ll_i)$ for $i = 1, \ldots, \ell$. Let $\circledcirc(\rho) = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$. A *configuration* of ρ is a $\leq$-closed subset of E. Let C be a configuration of ρ. A *C-subpath* of ρ is a maximal subpath $\varrho = n_u \xrightarrow{B_{u+1}} \ldots \xrightarrow{B_{u'}} n_{u'}$, such that $C \cap E_i \neq \emptyset$ for each $i = u, \ldots, u'$. For such a *C-subpath* ϱ, we define its *C-residue* to be the set $(E_{u+1} \cup E_{u+2} \cup \cdots \cup E_{u'}) - C$. Figure 10 illustrates these notions for a path $\rho = n_0 \xrightarrow{B_1} n_1 \xrightarrow{B_2} n_2 \xrightarrow{B_3} n_3 \xrightarrow{B_4} n_4 \xrightarrow{B_5} n_5 \xrightarrow{B_6} n_6 \xrightarrow{B_7} n_7$. Each causal MSC is represented by a rectangle. Events in the configuration C are indicated by small filled circles, events not in C but the C-residues are indicated by small blank circles, and events that are not in C nor in its residues are indicated by blank squares. Note that the configuration contains only events from B_1, B_3, B_4 and B_5. The two *C-subpaths* identified on Figure 10 are the sequences of transitions $\rho_1 = n_0 \xrightarrow{B_1} n_1$ and $\rho_2 = n_2 \xrightarrow{B_3} n_3 \xrightarrow{B_4} n_4 \xrightarrow{B_5} n_5$ that provide the events appearing in C. One can also notice from this example that *C-subpaths* do not depend on the length of a the considered path, and that the suffix of each path that does not contain an event in C can be ignored.

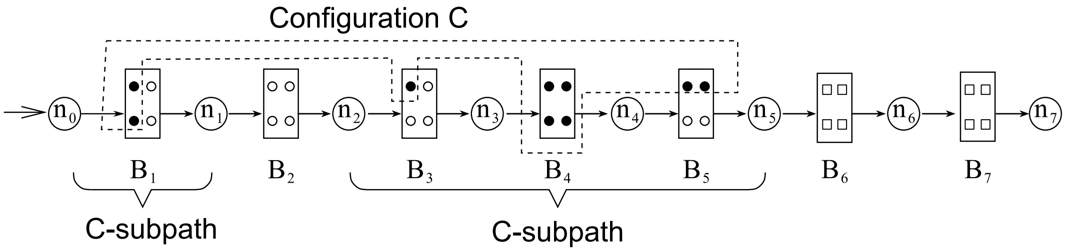

Fig. 10. An example of path, a configuration C, and its C-subpaths.

Lemma 2 *Let ρ be a path in H and C be a configuration of ρ. Then,*

(i) The number of C-subpaths of ρ is at most $K_{subpath} = |N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|}$.
(ii) Let ϱ be a C-subpath of ρ. Then the number of events in the C-residue of ϱ is at most $K_{residue} = |N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|} \cdot \max\{|B| \mid B \in \mathcal{B}\}$.

Proof:

(i) Suppose the contrary. Let $K = |\Sigma| \cdot 2^{|\mathcal{B}|}$. We can find $K+1$ C-subpaths whose ending nodes are equal. Let the indices of these $K+1$ ending nodes be $i_1 < i_2 < \ldots < i_{K+1}$. For $h = 1, \ldots, K$, let θ_h be the subpath of ρ from n_{i_h} to $n_{i_{h+1}}$; and let $\widehat{\mathcal{B}}_h$ be the set of causal MSCs appearing in θ_h. Hence we can find $\theta_{j_1}, \theta_{j_2}, \ldots, \theta_{j_{|\Sigma|}}$, $j_1 < j_2 < \ldots < j_{|\Sigma|}$, such that $\widehat{\mathcal{B}}_{j_1} = \widehat{\mathcal{B}}_{j_2} = \ldots = \widehat{\mathcal{B}}_{j_{|\Sigma|}}$. Pick an event e from $\circledcirc(\theta_{j_1})$ with $e \notin C$. Such an e exists, since, for example, none of the events in the first causal MSC appearing in θ_{j_1} is in C. Pick an event e' from $\circledcirc(\theta_{j_{|\Sigma|}})$ with $e' \in C$. Applying Lemma 1 yields that $e < e'$.

This leads to a contradiction, since C is $\leq$-closed.

(ii) Let $\varrho = n_i \xrightarrow{B_{i+1}} \ldots \xrightarrow{B_{i'}} n_{i'}$. Let $\widehat{E}_j = E_j - C$ for $j = i+1, \ldots, i'$. By similar arguments as in (i), it is easy to show that among $\widehat{E}_{i+1}$, …, $\widehat{E}_{i'}$, at most $|N| \cdot |\Sigma| \cdot 2^{|\mathcal{B}|}$ of them are nonempty. The claim then follows.

□

We are now ready to define the finite state automaton $\mathcal{A}_H = (S, S_{in}, \Sigma, S_{fi}, \Rightarrow)$ which accepts $Lin(H)$. As usual, S will be the set of states, $S_{in} \subseteq S$ the initial states, $\Longrightarrow \subseteq S \times \Sigma \times S$ the transition relation, and $S_{fi} \subseteq S$ the final states. Fix $K_{subpath}$, $K_{residue}$ to be the constants defined in Lemma 2. If $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ is a causal MSC and E' a subset of E, then we define the restriction of B to E' to be the causal MSC $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ as follows. As expected, λ' is the restriction of λ to E'; for each p, $\sqsubseteq'_p$ is the restriction of $\sqsubseteq_p$ to $(E' \cap E_p) \times (E' \cap E_p)$; and $\ll'$ is the restriction of $\ll$ to E'.

Intuitively, for a word σ in $\Sigma^\star$, $\mathcal{A}_H$ guesses an accepting path ρ of H and checks whether σ is in $Lin(\circledcirc(\rho))$. After reading a prefix σ' of σ, $\mathcal{A}_H$ memorizes a sequence of subpaths from which σ' was "linearized" (i.e the C-subpath of a path ρ such that C is a configuration reached after reading σ' and $\circledcirc(\rho)$ contains C). With Lemma 2, it will become clear later that at any time, we should remember at most $K_{subpath}$ such subpaths. Moreover, for each subpath, we need to know only a *bounded* amount of information, which will be stored in a data structure called "segment".

A causal MSC $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ is of size lower than K if $|E| \leq K$. A *segment* is a tuple (n, Γ, W, n'), where $n, n' \in N$, Γ is a nonempty subset of Σ, and W is either a non-empty causal MSC of size lower than $K_{residue}$, or the special symbol $\perp$. The state set S of $\mathcal{A}_H$ is the collection of finite sequences $\theta_1\theta_2 \ldots \theta_\ell$, $0 \leq \ell \leq K_{subpath}$, where each θ_i is a segment. Intuitively, a segment (n, Γ, W, n') keeps track of a subpath ϱ of H which starts at n and ends at n'. Γ is the collection of letters of events in $\circledcirc(\varrho)$ that have been "linearized". Finally, W is the restriction of $\circledcirc(\varrho)$ to the set of events in $\circledcirc(\varrho)$ that are not yet linearized. In case all events in $\circledcirc(\varrho)$ have been linearized, we set $W = \perp$. For convenience, we extend the operator $\circledcirc$ by: $W \circledcirc \perp = \perp \circledcirc W = W$ for any causal MSC W; and $\perp \circledcirc \perp = \perp$.

We define $\mathcal{A}_H = (S, S_{in}, \Sigma, S_{fi}, \Longrightarrow)$ as follows:

- As mentioned above, S is the collection of finite sequence of at most $K_{subpath}$ segments.
- The initial state set is $S_{in} = \{\varepsilon\}$, where ε is the null sequence.
- A state is final iff it consists of a single segment $\theta = (n, \Gamma, \perp, n')$ such that $n \in N_{in}$ and $n' \in N_{fi}$ (and Γ is any nonempty subset of Σ).
- The transition relation $\Longrightarrow$ of $\mathcal{A}_H$ is the least set satisfying the following

conditions.

—**Condition** (i):

Suppose $n \xrightarrow{B} n'$ where $B = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$. Let e be a minimal event in B (with respect to $\leq$) and let $a = \lambda(e)$. Let $\theta = (n, \Gamma, W, n')$ where $\Gamma = \{a\}$. Let $R = E - \{e\}$. If R is nonempty, then W is the restriction of B to R; otherwise we set $W = \perp$. Suppose $s = \theta_1 \ldots \theta_k \theta_{k+1} \ldots \theta_\ell$ is a state in S where $\theta_i = (n_i, \Gamma_i, W_i, n'_i)$ for each i. Suppose further that, e is a minimal event in $W_1 \circledcirc W_2 \circledcirc \ldots \circledcirc W_k \circledcirc W$.

- ("create a new segment") Let $\hat{s} = \theta_1 \ldots \theta_k \theta \theta_{k+1} \ldots \theta_\ell$. If $\hat{s}$ is in S, then $s \stackrel{a}{\Longrightarrow} \hat{s}$. In particular, for the initial state ε, we have $\varepsilon \stackrel{a}{\Longrightarrow} \theta$.
- ("add to the beginning of a segment") Suppose $n' = n_{k+1}$. Let $\hat{\theta} = (n, \Gamma \cup \Gamma_{k+1}, \widehat{W}, n'_{k+1})$, where $\widehat{W} = W \circledcirc W_{k+1}$. Let $\hat{s} = \theta_1 \ldots \theta_k \hat{\theta} \theta_{k+2} \ldots \theta_\ell$. If $\hat{s}$ is in S, then $s \stackrel{a}{\Longrightarrow} \hat{s}$.
- ("append to the end of a segment") Suppose $n = n'_k$. Let $\hat{\theta} = (n_k, \Gamma_k \cup \Gamma, \widehat{W}, n')$, where $\widehat{W} = W_k \circledcirc W$. Let $\hat{s} = \theta_1 \ldots \theta_{k-1} \hat{\theta} \theta_{k+1} \ldots \theta_\ell$. If $\hat{s}$ is in S, then $s \stackrel{a}{\Longrightarrow} \hat{s}$.
- ("glue two segments") Suppose $n = n'_k$ and $n' = n_{k+1}$. Let $\hat{\theta} = (n_k, \Gamma_k \cup \Gamma \cup \Gamma_{k+1}, \widehat{W}, n'_{k+1})$, where $\widehat{W} = W_k \circledcirc W \circledcirc W_{k+1}$. Let $\hat{s}$ be $\theta_1 \ldots \theta_{k-1} \hat{\theta} \theta_{k+2} \ldots \theta_\ell$. If $\hat{s}$ is in S, then $s \stackrel{a}{\Longrightarrow} \hat{s}$.

—**Condition** (ii):

Suppose $s = \theta_1 \ldots \theta_k \theta_{k+1} \ldots \theta_\ell$ is a state in S where $\theta_i = (n_i, \Gamma_i, W_i, n'_i)$ for $i = 1, 2, \ldots, \ell$. Suppose $W_k \neq \perp$. Let $W_k = (R_k, \eta_k, \{\sqsubseteq^k_p\}, \ll_k, \leq_k)$ and e a minimal event in W_k. Suppose further that e is a minimal event in $W_1 \circledcirc W_2 \circledcirc \ldots \circledcirc W_k$.

Let $\hat{\theta} = (n_k, \Gamma_k \cup \{a\}, \widehat{W}, n'_k)$, where $\widehat{W}$ is defined as follows. Let $\widehat{R} = R_k - \{e\}$. If $\widehat{R}$ is nonempty, then $\widehat{W}$ is the restriction of W to $\widehat{R}$; otherwise, set $\widehat{W} = \perp$. Let $\hat{s} = \theta_1 \ldots \theta_{k-1} \hat{\theta} \theta_{k+1} \ldots \theta_\ell$. Then we have $s \stackrel{a}{\Longrightarrow} \hat{s}$, where $a = \eta_k(e)$. (Note that $\hat{s}$ is guaranteed to be in S.)

We have now completed the construction of $\mathcal{A}_H$. It remains to show that $\mathcal{A}_H$ recognizes $Lin(H)$.

Lemma 3 *Let $\sigma \in \Sigma^\star$. Then σ is accepted by $\mathcal{A}_H$ iff σ is in $Lin(H)$.*

Proof: Let $\sigma = a_1 a_2 \ldots a_k$. Suppose σ is in $Lin(H)$. Let $\rho = n_0 \xrightarrow{B_1} \ldots \xrightarrow{B_\ell} n_\ell$ be an accepting path in H such that σ is a linearization of $\circledcirc(\rho)$. Hence we may suppose that $\circledcirc(\rho) = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$ where $E = \{e_1, e_2, \ldots, e_k\}$ and $\lambda(e_i) = a_i$ for $i = 1, \ldots, k$. And $e_i \leq e_j$ implies $i \leq j$ for any i, j in $\{1, \ldots, k\}$. Consider the configurations $C_i = \{e_1, e_2, \ldots, e_i\}$ for $i = 1, \ldots, k$. For each C_i, we can associate a state s_i in $\mathcal{A}_H$ as follows. Consider a fixed C_i. Let $\rho = \ldots \varrho_1 \ldots \varrho_2 \ldots \varrho_h \ldots$ where $\varrho_1, \varrho_2, \ldots, \varrho_h$ are the C_i-subpaths of ρ. Then we set $s_i = \theta_1 \ldots \theta_h$ where $\theta_j = (n_j, \Gamma_j, W_j, n'_j)$ with n_j being the starting node of ϱ_j, and Γ_j the collection of all $\lambda(e)$ for all events e that are in both $\circledcirc(\varrho_j)$

and C_i. Let R_j be the C_i-residue of ϱ_j. If R_j is nonempty, W_j is the causal MSC $(R_j, \eta_j, \{\sqsubseteq_p^j\}, \ll_j, \leq_j)$ where η_j is the restriction of λ to R_j; $\sqsubseteq_p^j$ is the restriction of $\sqsubseteq_p$ to those events in R_j that belong to process p, for each p; and $\ll_j$ the restriction of $\ll$ to R_j. If R_j is empty, then set $W_j = \perp$. Finally, n'_j is the ending node of ϱ_j.

Now it is routine (though tedious) to verify that $\varepsilon \xRightarrow{a_1} s_1 \ldots s_{k-1} \xRightarrow{a_k} s_k$ is an accepting run of $\mathcal{A}_H$. Conversely, given an accepting run of $\mathcal{A}_H$ over σ, it is straightforward to build a corresponding accepting path of H.

$\square$

With Lemma 3, we establish Theorem 2. As for complexity, the number of states in $\mathcal{A}_H$ is at most $(N_{seg})^{K_{subpath}}$, where N_{seg} is the maximal number of segments. Now, N_{seg} is $|N|^2 \cdot 2^{|\Sigma|} \cdot N_{res}$, where N_{res} is the possible number of residues. Recall that a residue is of size at most $K_{residue}$. According to Kleitman & Rotschild [17], the number of partial orders of size k is in $2^{f(k)}$ where $f(k) = \frac{1}{4}k^2 + \frac{3}{2}k + \mathcal{O}(\log_2(k))$. It follows that the number of $|\Sigma|$-labeled posets of size k is in $2^{f(k)} \cdot |\Sigma|^k$. All residues of size up to k can be encoded as a labeled poset of size k with useless events, labelled by a specific label $\sharp$. Hence the number of residues N_{res} is lower than $2^{f(K_{residue})} \cdot (|\Sigma| + 1)^{K_{residue}}$. Combining the above calculations then establishes the bound in theorem 2 on the number of states of $\mathcal{A}_H$.

3.3 Inclusion and Intersection of causal HMSC Languages

It is known that problems of inclusion, equality and non-emptiness of the intersection of the MSC languages associated with HMSCs are all undecidable [20]. Clearly, these undecidability results also apply to the causal languages of causal HMSCs. As in the case of HMSCs, theses problems are decidable for s-regular causal HMSCs since their linearization languages are regular.

It is natural to ask whether we can still obtain positive results for these problems beyond the subclass of s-regular causal HMSCs. In the setting of HMSCs, one can show that for all $K \geq 0$, the set of K-bounded linearizations of any globally-cooperative HMSC is regular. Moreover, for a suitable choice of K, the set of K-bounded linearizations is sufficient for effective verification [11]. Unfortunately, this result uses Kuske's encoding [18] into traces that is based on the existence of an (existential) bound on communication channels. Consequently, this technique does not apply to globally-cooperative causal HMSCs, as the visual language of a causal HMSC needs not be existentially bounded. For instance, consider the causal HMSC H of Figure 11. It is globally-cooperative and its visual language contains MSCs shown in the right part of

Figure 11: in order to receive the first message from p to r, the message from p to q and the message from q to r have to be sent and received. Hence every message from p to r has to be sent before receiving the first message from p to r, which means that H is not existentially bounded.

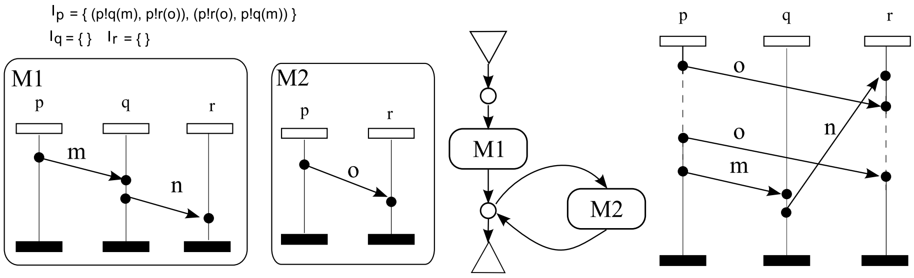

Fig. 11. A globally-cooperative causal HMSC that is not existentially bounded

We shall instead adapt the notion of atoms [1,15] and the techniques from [12].

Definition 6 *A causal MSC B is a* basic part *(w.r.t. the trace alphabets $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$) if there do not exist causal MSCs B_1, B_2 such that $B = B_1 \circledcirc B_2$.*

Note that we require that the set of events of a causal MSC is not empty. Now for a causal MSC B, we define a *decomposition* of B to be a sequence $B_1 \cdots B_\ell$ of basic parts such that $B = B_1 \circledcirc \cdots \circledcirc B_\ell$. For a set $\mathcal{B}$ of basic parts, we associate a trace alphabet $(\mathcal{B}, I_{\mathcal{B}})$ (w.r.t. the trace alphabets $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$) where $I_{\mathcal{B}}$ is given by: $B\ I_{\mathcal{B}}\ B'$ iff for every p, for every $\alpha \in Alph_p(B)$, for every $\alpha' \in Alph_p(B')$, it is the case that $\alpha\ I_p\ \alpha'$. We let $\sim_{\mathcal{B}}$ be the corresponding trace equivalence relation and denote the trace containing a sequence $u = B_1.\ldots.B_\ell$ in $\mathcal{B}^\star$ by $[u]_{\mathcal{B}}$ (or simply $[u]$). For a language $L \subseteq \mathcal{B}^\star$, we define its trace closure $[L]_{\mathcal{B}} = \bigcup_{u\in L} [u]_{\mathcal{B}}$. We begin by proving that the decomposition of any causal MSC B is unique up to commutation. More precisely,

Proposition 2 *Let $B_1 \ldots B_k$ be a decomposition of a causal MSC B. Then the set of decompositions of B is $[B_1 \ldots B_k]$.*

Proof: It is clear that every word in $[B_1 \ldots B_k]$ is a decomposition of B.

For the converse, let us suppose that there exists a decomposition $B = W_1 \circledcirc W_2 \circledcirc \cdots \circledcirc W_q$ such that $W_1 \ldots W_q \notin [B_1 \ldots B_k]$. It means that there exists a permutation ϕ and an index i with $W_j = B_{\phi(j)}$ for all $j < i$, $B_{\phi(1)} \cdots B_{\phi(k)} \in [B_1 \ldots B_k]$, and $W' = W_i \cap B_{\phi(i)} \neq \emptyset$ and $W'' = W_i \setminus B_{\phi(i)} \neq \emptyset$. It suffices now to prove that W' and W'' are causal MSCs and that $W_i = W' \circledcirc W''$ to get a contradiction with the fact that W_i is a basic part. By definition, the restriction of $\ll_{W_i}$ to W' is still a bijection (a send in W' matches a receive in W'). It implies that the restriction of $\ll_{W_i}$ to W'' is a bijection too. Both W' and W'' are thus causal MSCs. The fact that $W_i = W' \circledcirc W''$ comes from the definition of $\circledcirc$ and from the fact that $W' \subseteq B_{\phi(i)}$ and $W'' \subseteq B_{\phi(i+1)} \cdots B_{\phi(k)}$.

□

It is thus easy to compute the (finite) set of basic parts of a causal MSC B, denoted $Basic(B)$, since it suffices to find one of its decomposition.

Proposition 3 *For a given causal MSC B, we can effectively decompose B in time $O(|B|^2)$.*

Proof: Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$. We describe the decomposition of B, which is analogous to the technique in [15]. We consider the *directed* graph $(E, \leq \cup R)$, where R is the symmetric closure of $\ll \cup \left(\bigcup_{p \in \mathcal{P}} R'_p \cup R''_p\right)$, with

$$R'_p = \{(e, e') \in E_p \times E_p \mid e \widehat{\sqsubseteq}_p e' \text{ and } \lambda(e) \ I_p \ \lambda(e')\} \ ,$$
$$R''_p = \{(e, e') \in E_p \times E_p \mid e \not\sqsubseteq_p e' \text{ and } e' \not\sqsubseteq_p e \text{ and } \lambda(e) \ D_p \ \lambda(e')\} \ .$$

Intuitively, R'_p denote pairs of events that are ordered in B, but which labels are independent in I_p. As this ordering can not be obtained via composition, this ordering should appear in the decomposition of B, that is e and e' should belong to the same basic part. Similarly, relation R''_p contains pairs of events that are unordered in B, but which labels are dependent.

For each strongly connected component E' of $(E, \leq \cup R)$, we associate a structure $C = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$, where λ' is the restriction of λ to E', $\sqsubseteq'_p$ is the restriction of $\sqsubseteq'_p$ to E', and $\ll'$ is the restriction of $\ll$ to E'. It is easy to see that C is a causal MSC, since each receive needs to be in the same strongly connected component than its associated send (since the relation includes the symmetric closure of $\ll$). We first prove that C is a basic part. By contradiction, otherwise, we would have $C = B_1 \circledcirc B_2$, which by definition of $\circledcirc$ means that no edge of $\leq \cup R$ can go from one event of B_2 to one event of B_1, which contradicts the fact that E' is strongly connected.

Let $E_1, \ldots, E_n$ be the set of basic parts obtained. We order them such that there is no edge of $\leq \cup R$ from any event of E_j to some event of E_i with $i < j$ (it is always possible, else E_i, E_j would be in the same strongly connected component). It is now clear that $B = E_1 \circledcirc \cdots \circledcirc E_n$, since no event of E_j can be before an event of E_i, $i < j$ (else there would be an edge of $\leq$ from E_j to E_i). Notice that the decomposition in strongly connected components with Tarjan's Algorithm is in linear time in the number of edges, that is linear in $|B| + \sum_{p \in \mathcal{P}} | \sqsubseteq_p | \leq |B| + |B|^2$, where $|B|$ is the number of events of B. For comparison, recall that the complexity of decomposing an MSC B in atoms [15] is in $O(2|B|)$ (the immediate successor relation in MSCs is the union of the message pairing relation and the total ordering on instances, that is there are at most 2 immediate successors for a given event). □

In view of Proposition 3, we assume through the rest of this section that every transition of a causal HMSC H is labelled by a basic part. This obviously incurs no loss of generality, since we can simply decompose each causal MSC in H into basic parts and decompose any transition of H into a sequence of transitions labeled by these basic parts. Given a causal HMSC H, we let $Basic(H)$ be the set of basic parts labelling transitions of H. Trivially, a causal MSC is uniquely defined by its basic part decomposition. Then instead of the visual language we can use the *basic part language* of H, denoted by $BP(H) = \{B_1 \dots B_\ell \in Basic(H)^\star \mid B_1 \circledcirc \dots \circledcirc B_\ell \in cMSC(H)\}$. Notice that $BP(H) = [BP(H)]$ by Proposition 2, that is, $BP(H)$ is closed by commutation. We can also view H as a finite state automaton over the alphabet $Basic(H)$, and denote by $\mathcal{L}_{Basic}(H) = \{B_1 \cdots B_\ell \in Basic(H)^\star \mid n_0 \xrightarrow{B_1} n_1 \cdots \xrightarrow{B_\ell} n_\ell$ is an accepting path of H$\}$ its associated (regular) language. We now relate $BP(H)$ and $\mathcal{L}_{Basic}(H)$.

Proposition 4 *Let H be a causal HMSC. Then $BP(H) = [\mathcal{L}_{Basic}(H)]$.*

Proof: First, let us take a word w in $[\mathcal{L}_{Basic}(H)]$. Thus $w = B_1 \dots B_k \sim B_{i_1} \dots B_{i_k}$ such that $B_{i_1} \dots B_{i_k} \in \mathcal{L}_{Basic}(H)$. As $\mathcal{L}_{Basic}(H) \subseteq BP(H)$ and $B_1 \circledcirc \dots \circledcirc B_k = B_{i_1} \circledcirc \dots \circledcirc B_{i_k}$ we conclude that $[\mathcal{L}_{Basic}(H)] \subseteq BP(H)$. Second, let us take a word w in $BP(H)$. Let us note $\circledcirc(w)$ its corresponding causal MSC, i.e. for $w = B_1 \dots B_k$, $\circledcirc(w) = B_1 \circledcirc \dots \circledcirc B_k$. Then this word is generated by an accepting path $\rho = n_0 \xrightarrow{P_1} n_1 \dots \xrightarrow{P_l} n_l$ of H such that $\circledcirc(w) = P_1 \circledcirc \dots \circledcirc P_l$. By proposition 2, we know that any other decomposition of $\circledcirc(w)$ belongs to $[B_1 \dots B_k]$, and in particular, the one we choose. Thus we obtain that $BP(H) \subseteq [\mathcal{L}_{Basic}(H)]$. □

Assuming we know how to compute the trace closure of the regular language $\mathcal{L}_{Basic}(H)$, we can obtain $BP(H)$ with the help of Proposition 4. In general, we cannot effectively compute this language. However if H is globally-cooperative, then $[\mathcal{L}_{Basic}(H)]$ is regular and a finite state automaton recognizing $[\mathcal{L}_{Basic}(H)]$ can be effectively constructed [8,20]. Considering globally-cooperative causal HMSCs as finite state automata over basic parts, we can apply [20] to obtain the following decidability and complexity results:

Theorem 3 *Let H, H' be causal HMSCs over the* same *family of trace alphabets $\{(\Sigma_p, I_p)\}_{p\in\mathcal{P}}$. Suppose H' is globally-cooperative. Then we can build a finite state automaton $\mathcal{A}'$ over $Basic(H')$ such that $\mathcal{L}_{Basic}(A') = [\mathcal{L}_{Basic}(H')]$. Moreover, $\mathcal{A}'$ has at most $2^{O(n\cdot b)}$ states, where n is the number of nodes in H and b is the number of basic parts in $Basic(H)$. Consequently, the following problems are decidable:*

(i) Is $cMSC(H) \subseteq cMSC(H')$?
(ii) Is $cMSC(H) \cap cMSC(H') = \emptyset$?

Furthermore, the complexity of (i) is PSPACE-complete and that of (ii) is EXPSPACE-complete.

The above theorem shows that we can model-check a causal HMSC against a globally-cooperative causal HMSC specification. Note that we can only apply Theorem 3 to two causal HMSCs over the *same* family of trace alphabets. If the causal HMSCs H, H' in theorem 3 satisfy the additional condition that every causal MSC labeling the transitions of H and H' respects $\{(\Sigma_p, I_p)\}_{p \in \mathcal{P}}$, then we can compare the visual languages $Vis(H)$ and $Vis(H')$, thanks to Proposition 1.

On the other hand, if the independence relations are different, the atoms of H and H' are unrelated. Theorem 4 proves that comparing the MSC languages of two globally-cooperative causal HMSCs H, H' using different independence relations is actually undecidable. The only way to compare them is then to compare their linearization languages. Consequently, we would need to work with s-regular causal HMSCs.

Theorem 4 *Let G, H be globally-cooperative causal HMSCs with respectively families of trace alphabets $\{(\Sigma_p, I_p)\}_{p \in \mathcal{P}}$ and $\{(\Sigma_p, J_p)\}_{p \in \mathcal{P}}$, where for each p, I_p and J_p are allowed to differ. Then it is undecidable to determine whether $Vis(G) \cap Vis(H) = \emptyset$.*

Proof: We reduce the PCP problem, which is well known to be undecidable, to the emptiness of the intersection of the visual languages of two (globally-cooperative) causal HMSCs, if we *do not* assume that both causal HMSCs use the same independence relation.

Let J be a finite set and $(v_i, w_i)_{i \in J}$ be an instance of PCP, with $v_i, w_i \in \{a, b\}^* \setminus \epsilon$ for all $i \in J$. We will use two causal HMSCs H_1 and H_2 to encode the PCP. The intuition for the reduction is that the causal HMSC H_1 generates sequences of CaMSCs of the form $(V_i W_i)^*$, where CaMSCs V_i and W_i represent respectively the words v_i and w_i. The causal HMSC H_2 generates sequences of the form $(A\bar{A} \vee B\bar{B})^*$, where the causal MSCs X and $\bar{X}$ represent an x letter in v and w respectively, with $x \in \{a, b\}$.

We have three process, P_1, P_2 and P_3. The causal MSCs A, B are made of a single message from process P_1 to process P_3, respectively labeled by a and b. The causal MSCs $\bar{A}, \bar{B}$ are made of two messages both labeled by the same letter (respectively $\bar{a}$ and $\bar{b}$). The first message is from process P_1 to P_2, and the second message is sent after the reception of the first message, from process P_2 to process P_3. For each pair (v_i, w_i) in the PCP instance, we build two causal MSCs V_i and W_i. If v_i is of the form $v_i = xyz$, the causal MSC V_i is made of the concatenation of X, Y, Z. Similarly, W_i is made of the concatenation $\bar{X}, \bar{Y}, \bar{Z}$, when $w_i = xyz$. These causal MSCs are depicted in Figure 12.

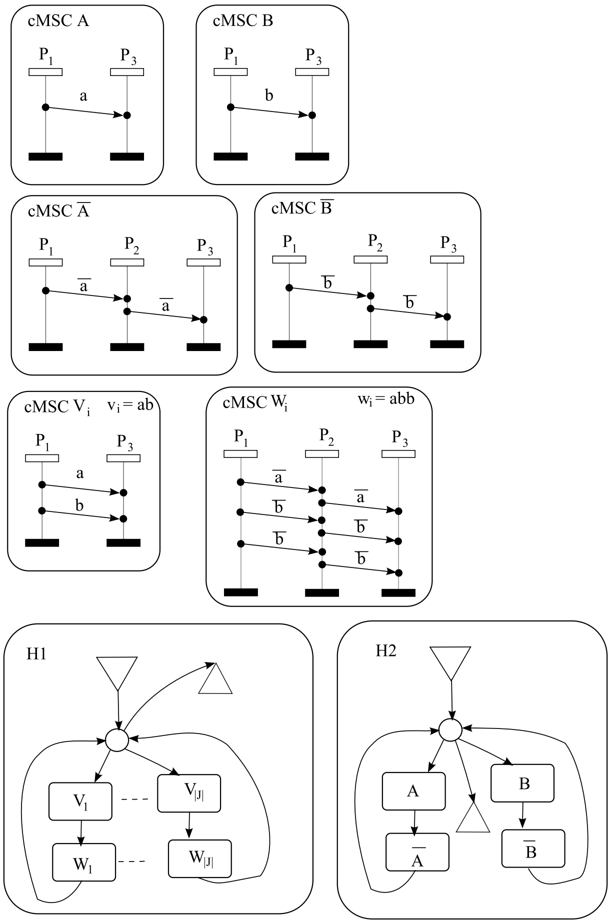

Fig. 12. PCP encoding with two causal HMSCs

Let us denote by V the labels appearing in V_i's and by W the labels appearing in W_i's. The independence relation I_1 for H_1 states that all events on process P_2 and P_3 commute. On process P_1, events labeled by a letter of V (namely $P_1!P_3(a)$ and $P_1!P_3(b)$ commute with events labeled by a letter of W (namely $P_1!P_2(\bar{a})$ and $P_1!P_2(\bar{b})$. There is no commutation among events labeled by a letter of V, and no commutations among events labeled by a letter of W. In the same way, the independence relation I_2 for H_2 states that no events on process 1 and 2 commute. On process 3, events from v (namely $3?1(a)$ and $3?1(b)$ commute with events from w (namely $3?2(\bar{a})$ and $3?2(\bar{b})$. There is no commutations among events from v, and no commutations among events from w.

It is easy to check that both H_1 and H_2 are globally-cooperative. Indeed, notice first that the letters a and b behave exactly the same. We can then forget about them for global cooperativeness, and draw the communication graph considering only 6 letters $P_1!P_2, P_2?P_1, P_2!P_3, P_3?P_1, P_1!P_3, P_3?P_1$. Every elementary cycle of H_1 contains a V_i and a W_i. Since v_i, w_i are non empty words, every of these 6 letters appear in every loop of H_1. In particular, we have the undirected relation $P_1!P_2 - P_2?P_1 - P_2!P_3 - P_3?P_2 - P_3?P_1 - P_1!P_3$, which proves that the graph is connected. In the same way, every loop of H_2 contains a $X\bar{X}$, hence every of the 6 letters appear in every elementary cycle (and loop) of H_2. This time, the graph is connected, but through another undirected path: $P_3?P_1 - P_1!P_3 - P_1!P_2 - P_2?P_1 - P_2!P_3 - P_3?P_2$. Hence, both H_1 and H_2 are globally-cooperative.

Assume that $Vis(H_1) \cap Vis(H_2) \neq \emptyset$. Let $M \in Vis(H_1) \cap Vis(H_2)$. Let v be the projection of M on alphabet $P_1!P_3(a), P_1!P_3(b)$, and w the projection of M on alphabet $P_1!P_2(\bar{a}), P_1!P_2(\bar{b})$. Now, because $M \in Vis(H_2)$ and since there is no commutation on process P_1 allowed by I_2, we get that $v = w$, confusing $P_1!P_2(\bar{a})$ with $P_1!P_3(a)$ and $P_1!P_2(\bar{b})$ with $P_1!P_3(b)$.

Second, because $M \in Vis(H_1)$, there exists a sequence $i_1 \cdots i_n \in J^*$ with $M \in Vis(V_{i_1} \circledcirc W_{i_1} \circledcirc \cdots V_{i_n} \circledcirc W_{i_n})$. Since by I_1, there is no commutation among letters of v, the projection v of M on alphabet $P_1!P_3(a), P_1!P_3(b)$ is the same as the projection of $V_{i_1} \circledcirc W_{i_1} \circledcirc \cdots V_{i_n} \circledcirc W_{i_n}$. That is, $v = v_{i_1} \cdots v_{i_n}$ (confusing letter a with $P_1!P_3(a)$ and letter b with $P_1!P_3(b)$). In the same way, $w = w_{i_1} \cdots w_{i_n}$ (confusing letter a with $P_1!P_2(\bar{a})$ and letter b with $P_1!P_2(\bar{b})$). That is $v_{i_1} \cdots v_{i_n} = v = w = w_{i_1} \cdots w_{i_n}$, which proves that it is a solution for the PCP problem.

Now, assume that the instance $(v_i, w_i)_{i \in I}$ of PCP has a solution $v_{i_1} \cdots v_{i_n} = w_{i_1} \cdots w_{i_n} = x_1 \cdots x_m$. Consider the following MSC M. We describe M process by process (which is enough to uniquely define M since all MSCs in $Vis(H_1)$ and $Vis(H_2)$ are weak FIFO). On process P_1, M is of the form $P_1!P_3(x_1)P_1!P_2(\bar{x_1}) \cdots P_1!P_3(x_m)P_1!P_2(\bar{x_m})$. On process P_2, M is of the form $P_2?P_1(\bar{x_1})P_2!P_3(\bar{x_1}) \cdots P_2?P_1(\bar{x_m})P_2!P_3(\bar{x_m})$. On process 3, M is of the form $P_3?P_1(a_1)P_3?P_1(b_1)P_3?P_1(c_1)P_3?P_2(\bar{d_1})P_3?P_2(\bar{e_1})P_3?P_1(\bar{f_1}) \cdots P_3?P_1(a_n)$ $P_3?P_1(b_n)P_3?P_1(c_n)P_3?P_2(\bar{d_n})P_3?P_2(\bar{e_n})P_3?P_1(\bar{f_n})$, where for all j, $v_{i_j} = a_j b_j c_j$ and $w_{i_j} = d_j e_j f_j$. It is easy to see that $M \in Vis(H_1) \cap Vis(H_2)$, which ends the proof. $\square$

4 Window-bounded causal HMSCs

One of the chief attractions of causal MSCs is that they enable the specification of behaviors containing braids of arbitrary size such as those generated by sliding windows protocols. Very often, sliding windows protocols appear in a situation where two processes p and q exchange bidirectional data. Messages from p to q are of course used to transfer information, but also to acknowledge messages from q to p. If we abstract the type of messages exchanged, these protocols can be seen as a series of query messages from p to q and answer messages from q to p. Implementing a sliding window means that a process may send several queries in advance without needing to wait for an answer to each query before sending the next query. Very often, these mechanisms tolerate losses, i.e. the information sent is stored locally, and can be retransmitted if needed (as in the alternating bit protocol). To avoid memory leaks, the number of messages that can be sent in advance is often bounded by some integer k, that is called the size of the sliding window. Note however that for scenario languages defined using causal HMSCs, such window sizes do not always exist. This is the case for example for the causal HMSC depicted in Figure 1 with independence relations $I_p = \{((p!q(Q), p?q(A)), (p?q(A), p!q(Q)))\}$ and $I_q = \{((q?p(Q), q!p(A)), (q!p(A), q?p(Q))\}$. The language generated by this causal HMSC contains scenarios where an arbitrary number of messages from p to q can cross an arbitrary number of messages from q to p. A question that naturally arises is to know if the number of messages crossings is bounded by some constant in all the executions of a protocol specified by a causal HMSC. In what follows, we define these crossings, and show that their boundedness is a decidable problem.

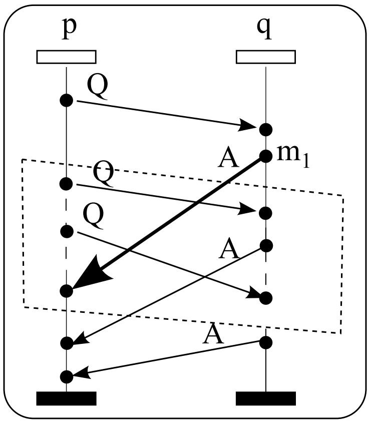

Fig. 13. Window of message m_1

Definition 7 *Let $M = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ be an MSC For a message (e, f) in M, that is, $(e, f) \in \ll$, we define the* window *of (e, f), denoted $W_M(e, f)$, as the set of messages $\{e' \ll f' \mid loc(\lambda(e')) = loc(\lambda(f))$ and $loc(\lambda(f')) = loc(\lambda(e))$ and $e \leq f'$ and $e' \leq f\}$.*

We say that a causal HMSC H is *K-window-bounded* iff for every $M \in Vis(H)$ and for every message (e, f) of M, it is the case that $|W_M(e, f)| \leq K$. H is said to be window-bounded iff H is K-window-bounded for some K.

Figure 13 illustrates notion of window, where the window of the message m_1 (the first answer from q to p) is symbolized by the area delimited by dotted lines. It consists of all but the first message Q from p to q. Clearly, the causal HMSC H of Figure 1 is not window-bounded. We now describe an algorithm to effectively check whether a causal HMSC is window bounded. It builds a finite state automaton whose states remember the labels of events that must appear in the *future* of messages (respectively in the *past*) in any MSC of $Vis(H)$.

Formally, for a causal MSC $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ and $(e, f) \in \ll$ a message of B, we define the future and past of (e, f) in B as follows:

$$Future_B(e, f) = \{a \in \Sigma \mid \exists x \in E, f \leq x \wedge \lambda(x) = a\}$$
$$Past_B(e, f) = \{a \in \Sigma \mid \exists x \in E, x \leq e \wedge \lambda(x) = a\}$$

Notice that for a message $m = (e, f)$, we always have $\lambda(e) \in Past_B(m)$ and $\lambda(f) \in Future_B(m)$. For instance, in Figure 13, $Past_B(m_1) = \{p!q(Q), q?p(Q), q!p(A)\}$.

Intuitively, if a letter of the form $p!q(m)$ is in the future of a message (e, f) in a causal MSC B, then any occurrence of message m that is appended to B *is* in the future of (e, f). Hence, this message can not appear in the window of (e, f). Note that a symmetric property holds for the past of (e, f).

Proposition 5 *Let $B = (E, \lambda, \{\sqsubseteq_p\}, \ll)$ and $B' = (E', \lambda', \{\sqsubseteq'_p\}, \ll')$ be two causal MSCs, and let $m \in \ll$ be a message of B. Then we have:*

$$Future_{B \circledcirc B'}(m) = Future_B(m) \cup \{a' \in \Sigma \mid \exists x, y \in E'$$
$$\exists a \in Future_B(m) \text{ s.t. } \lambda(y) = a' \wedge x \leq' y \wedge a \; D_{loc(a)} \; \lambda(x)\}$$

$$Past_{B' \circledcirc B}(m) = Past_B(m) \cup \{a' \in \Sigma \mid \exists x, y \in E'$$
$$\exists a \in Future_B(m) \text{ s.t. } \lambda(y) = a' \wedge y \leq' x \wedge a \; D_{loc(a)} \; \lambda(x)\}$$

Proof: Follows from definition. □

Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a causal HMSC. Consider a path ρ of H with $\circledcirc(\rho) = B_1 \circledcirc \cdots \circledcirc B_\ell$ and a message m in B_1. First, the sequence of sets $Future_{B_1}(m)$, $Future_{B_1 \circledcirc B_2}(m)$, $\ldots$, $Future_{B_1 \circledcirc \cdots \circledcirc B_\ell}(m)$ is non-decreasing. Using proposition 5, these sets can be computed on the fly and with a finite number of states. Similar arguments hold for the past sets. Now consider a message (e, f) in a causal MSC B labelling some transition t of H. With the above observation on *Future* and *Past*, we can show that, if there is a bound $K_{(e,f)}$ such that the window of a message (e, f) in the causal MSC generated by any path containing t is bounded by $K_{(e,f)}$, then $K_{(e,f)}$ is at most $b|N||\Sigma|$ where $b = \max\{|B| \mid B \in \mathcal{B}\}$. Further, we can effectively determine whether such a bound $K_{(e,f)}$ exists by constructing a finite state transition system whose states memorize the future and past of (e, f). Thus we have the following:

Theorem 5 *Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a causal HMSC. Then we have:*

(i) If H is window-bounded, then H is K-window-bounded, where K is at most $b \cdot |N| \cdot |\Sigma|$.
(ii) Further, we can effectively determine whether H is window-bounded in time $O(s \cdot |N|^2 \cdot 2^{|\Sigma|})$, where s is the sum of the sizes of causal MSCs in $\mathcal{B}$.

The rest of this subsection is devoted to the proof of Theorem 5. We fix H as in Theorem 5.

Proof: [of Theorem 5(i)] Suppose that H is not k-window-bounded, where $k = b \cdot |N| \cdot |\Sigma|$. Let $B \in \mathcal{B}$ be a causal MSC in $cMSC(H)$, and let the pair (e, f) be a message of B. Let ρ be a path of H, and $V \in Vis(\circledcirc(\rho))$ be an MSC such that the message (e, f) is crossed by $k + 1$ messages in V. Recall that a causal MSC contains at most $b/2$ messages. Then, ρ contains at least $2|\Sigma|$ occurrences of a node n', such that the label of n' contains at least a message m' that crosses (e, f) in V. Without loss of generality, we can consider that n' is repeated at least $|\Sigma|$ times after B in ρ (else we apply a symmetric proof, considering repetitions of B' occurring before B). That is, ρ is of the form $\cdots n_0 \rightarrow \cdots n_1 \rightarrow \cdots n_2 \rightarrow \cdots n_{|\Sigma|} \rightarrow \cdots$, where $n_1 = \cdots = n_{|\Sigma|} = n'$.

Let us denote by E_i the label of the prefix $\cdots n_0 \rightarrow \cdots n_1 \rightarrow \cdots n_2 \rightarrow \cdots n_i$ of ρ, for $i = 0, 1, 2, \ldots, |\Sigma|$. Consider the sequence of sets $F_i = Future_{E_i}(e, f)$, $i = 0, 1, \ldots, |\Sigma|$. Each F_i is a subset of Σ and the sequence F_0, F_1, $\ldots$, $F_{|\Sigma|}$ is non-decreasing. Hence, we can find $\ell \leq |\Sigma|$ such that $F_\ell = F_{\ell+1}$. This means that the path ρ', which is computed from path ρ by repeating twice the cycle between n_ℓ and $n_{\ell+1}$ (n_ℓ excluded but $n_{\ell+1}$ included), have at least one more message m' which crosses m. Thus, we can exhibit a new execution $V' \in Vis(\circledcirc(\rho'))$ such that (e, f) is crossed by at least $k + 2$ messages. As we can iterate this construction, it means that H is not window-bounded.

We next establish Theorem 5(ii). We shall show that for a given message m, one can decide in an efficient way whether there is a window bound, by constructing a finite state transition system that memorizes $Future(m)$ and $Past(m)$.

For a given causal HMSC $H = (N, N_{in}, \mathcal{B}, N_{fi}, \longrightarrow)$ and a message (e, f) of some causal MSC $B \in \mathcal{B}$, we build the following transition system $\mathcal{A}_{(e,f)} = (Q, Q_{in}, \mathcal{B}, Q_{fi}, \delta)$ that computes the possible futures of (e, f), where:

- $Q = N \times 2^{\Sigma}$ is a set of states, recalling a node in H and a future,
- $Q_{in} = \{(n, \emptyset) \mid n \in N_{in}\}$ is the set of initial states,
- $(n, X) \in Q_{fi}$ if and only if $n \in N_{fi}$.
- $\delta \subseteq Q \times \mathcal{B} \times Q$ is the least transition relation such that:
  · $\big((n, \emptyset), B, (n', \emptyset)\big) \in \delta$ if $n \xrightarrow{B} n'$

- $\big((n,\emptyset), B, (n', \mathit{Future}_B(e,f))\big) \in \delta$ if $n \xrightarrow{B} n'$ and (e,f) belongs to B.
- $\big((n,X), B, (n', X')\big) \in \delta$, where $B = (E, \lambda, \{\sqsubseteq_p\}, \ll, \leq)$, if $n \xrightarrow{B} n'$, and $X' = X \cup \{\lambda(y) \mid \exists x, y \in E, \exists a \in X \wedge a \; D \; \lambda(x) \wedge x \leq y\}$.

Note that the first rule in the construction of the transition relations of $\mathcal{A}_{(e,f)}$ simply copies the transitions of H. The second rule perform a random choice of a message in a random occurrence of a transition of H labelled by B. This rule is important, as it allows to chose nondeterministically an occurrence (e,f) of a message after an arbitrary path in the HMSC. The last rule updates the futures or pasts after the choice of a message occurrence. A state $q = (n, X)$ in $\mathcal{A}_{(e,f)}$ represents a possible set X of labels in $\mathit{Future}_{\circledcirc(\rho)}(e,f)$ for some path ρ that ends (respectively starts) at node n in H, and contains a message (e,f). Slightly abusing the notation, we will denote by $\mathit{Future}(q)$ (resp. $\mathit{Past}(q)$) the set X. Note that in any strongly connected subset $C = \{q_1, \ldots, q_k\}$ of $\mathcal{A}_{(e,f)}$ (respectively $\mathcal{A}'_{(e,f)}$), $\mathit{Future}(q_1) = \mathit{Future}(q_2) = \cdots = \mathit{Future}(q_k)$ (resp. $\mathit{Past}(q_1) = \mathit{Past}(q_2) = \cdots = \mathit{Past}(q_k)$). Hence, we will denote by $\mathit{Future}(C)$ (resp. $\mathit{Past}(C)$) the set of observed labels on any state of C.

We can also build a finite state transition system $\mathcal{A}'_{(e,f)}$ that computes the possible pasts of (e,f), by a backward search in the causal HMSC H.

We observe the following properties of the finite state automata $\mathcal{A}_{(e,f)}$ and $\mathcal{A}'_{(e,f)}$.

Lemma 4 *Let $H = (N, N_{in}, \mathcal{B}, \longrightarrow, N_{fi})$ be a causal HMSC. Let B be a causal MSC in $\mathcal{B}$ and (e,f) a message in B with the label of e being $p!q(m)$. Consider the finite state automata $\mathcal{A}_{(e,f)}$ and $\mathcal{A}'_{(e,f)}$ as constructed above. Then, H is window-bounded iff both of the following conditions hold:*

- *There does* not *exist a strongly connected component C in $\mathcal{A}_{(e,f)}$ and a letter $q!p(m') \in \Sigma$ such that $q!p(m')$ is in $\mathit{Alph}(B) - \mathit{Future}(C)$ for some causal MSC B labelling a transition in C.*
- *There does* not *exist a strongly connected component C in $\mathcal{A}'_{(e,f)}$ and a letter $q!p(m') \in \Sigma$ such that $q!p(m')$ is in $\mathit{Alph}(B) - \mathit{Past}(C)$ for some causal MSC B labelling a transition in C.*

Proof: One direction is straightforward. If any of these strongly connected components exists (either before or after m), then there is an unbounded number of path generating an unbounded number of occurrences of $q!p(m')$ that are not causally related to m. Hence, for each of these path, there is a visual extension where all m' generated by occurrences of the cycle cross m, and the window size of m is not bounded. The other direction is a direct consequence of Theorem 5(i). □

Thus Theorem 5(ii) follows from Lemma 4. It remains to establish the complexity claim in Theorem 5(ii). The transition system $\mathcal{A}_{(e,f)}$ has at most $|N| \times 2^{|\Sigma|}$ states, and we have to analyze strongly connected components of $\mathcal{A}_{(e,f)}$. However, as noticed before, every strongly connected component of $\mathcal{A}_{(e,f)}$ enjoys the property to have a second component which is constant. Hence we need to test the property only for *maximal* strongly connected components. Indeed, if C is a strongly connected component of $\mathcal{A}_{(e,f)}$ such that $q!p(m')$ is the label of an event in a causal MSC labeling a transition of C but that is not in $\mathit{Future}(C)$, then we can consider the maximal strongly connected component D of $\mathcal{A}_{(e,f)}$ containing C (it exists since the union of two non disjoint strongly connected components is again a strongly connected component). Since D is a strongly connected component, its second component $\mathit{Future}(D)$ is constant, hence $\mathit{Future}(D) = \mathit{Future}(C)$. Since $C \subseteq D$, we have that $q!p(m')$ is a label of an event of D and is not in $\mathit{Future}(C) = \mathit{Future}(D)$.

Using Tarjan's algorithm [26], we can compute in quadratic time the partition of $\mathcal{A}_{(e,f)}$ into maximal strongly connected components (for each set $X \subseteq 2^{\Sigma}$, we partition the subpart of $\mathcal{A}_{(e,f)}$ with a constant second component being X). Then for each maximal strongly connected component (C, X), it suffices to compute $\lambda(C)$ and to compare it with X, which is linear in n. Hence, the overall complexity of the algorithm is in $O(|N|^2 \cdot 2^{|\Sigma|})$. As, we build the two automata $\mathcal{A}_{(e,f)}$ and $\mathcal{A}_{(e,f)}$ for each occurrence (e, f) of a message in each causal MSC labeling a transition of H, we obtain a complexity in $\mathcal{O}(s.|N|^2.2^{|\Sigma|})$, where s is the sum of the sizes of causal MSCs in $\mathcal{B}$. $\square$

5 Relationship with Other Scenario Models

We compare here the expressive power of other HMSC-based scenario languages with causal HMSCs. For comparison, we will only consider weak-FIFO scenario languages, that is HMSCs that are labelled by weak FIFO MSCs and causal HMSCs that are labelled by causal MSCs which visual extensions are weak FIFO. Indeed, non weak-FIFO scenarios can be seen as a little degenerate descriptions, as they can be differentiated by their visual languages, but not by their linearization languages. Hence, within this weak-FIFO setting, the comparisons established in this section holds both for visual languages and linearization languages.

The first topic of comparison is causal HMSCs themselves. Indeed, a previous definition [10] of s-regular and globally-cooperative causal HMSCs required that for every $p \in \mathcal{P}$, and for every B labeling a cycle of a causal HMSC, $\mathit{Alph}_p(B)$ was D_p-connected. It is not necessary in the definition of this paper: two letters from the same process can be connected through communication via another process, and not directly on the same process. It is important in the following setting.

An important question is the class of Communicating Finite State Machine (CFM for short) corresponding to HMSC languages. It has been shown in [14] that s-regular (compositional) HMSCs corresponds to universally bounded CFMs. The natural model to compare causal HMSCs and CFMs could be asynchronous cellular automata with type [4], also called mixed machine in [9], which allows communication both through Mazurkiewicz traces and messages. It has been shown in [9] that using the same s-regular definition as in this paper, universally bounded mixed model and s-regular causal (compositional) HMSCs coincide. It is easy to see that this is not the case with the old definition of [10]. Consider the following example: a causal HMSC composed of a single loop labelled by a causal MSC M that contains four unordered messages:m, o from process p to process q, and n, g from process q to process p. The dependence relation D_p is defined as the reflexive and symmetric closure of $\{(p!q(m), p?q(g)); (p?q(n), p!q(o))\}$ and $D_q = \Sigma_q \times \Sigma_q$. This example and the communication graph associated to its single loop are represented in Figure 14. Clearly, this causal HMSC is not s-regular nor even globally-cooperative with the definition of [10] (Σ_p is not D_p-connected), but it is s-regular with the definition of this paper. Also, there is no globally-cooperative causal HMSC in the terms of [10] recognizing the same language.

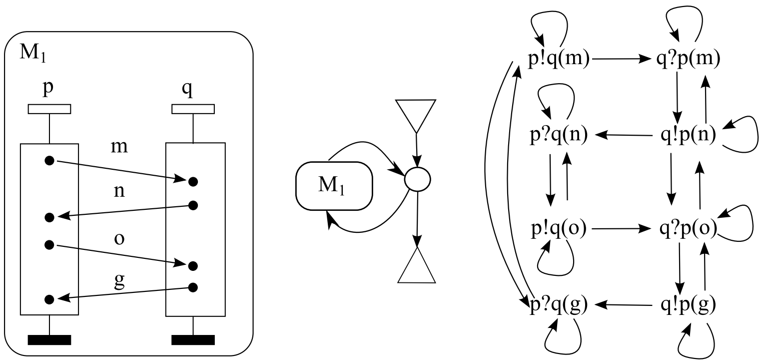

Fig. 14. A s-regular causal HMSC with the communication graph of its cycle

Then, we consider HMSCs. Two important strict HMSC subclasses are (i) *s-regular* (also called regular in [20] and bounded in [3]) HMSCs which ensure that the linearizations form a regular set and (ii) *globally-cooperative* HMSCs [12], which ensure that for a suitable choice of K, the set of K-bounded linearizations form a regular representative set. By definition, causal HMSCs, s-regular causal HMSCs and globally-cooperative causal HMSCs extend respectively HMSCs, s-regular HMSCs and globally-cooperative HMSCs.

Figure 11 shows a globally-cooperative causal HMSC which is not in the subclass of s-regular causal HMSCs. Thus, s-regular causal HMSCs form a strict subclass of globally-cooperative causal HMSCs. Trivially, globally-cooperative causal HMSCs are a strict subclass of causal HMSCs. Figure 9 displays a s-regular causal HMSC whose visual language is not finitely generated. It follows that (s-regular/globally-cooperative) causal HMSCs are strictly more powerful than (s-regular/globally-cooperative) HMSCs.

Another extension of HMSCs is *compositional* HMSCs (or CHMSCs for short), first introduced by [13]. CHMSCs generalize HMSCs by allowing dangling message-sending and message-reception events, i.e. the message pairing relation $\ll$ in a compositional MSC is only a partial non-surjective mapping contained in $E_! \times E_?$. The concatenation of two compositional MSCs $M \circ M'$ performs the instance-wise concatenation as for MSCs, and computes a new message pairing relation $\ll''$ defined over $(E_! \cup E'_!) \times (E_? \cup E'_?)$ extending $\ll \cup \ll'$, and preserving the FIFO ordering of messages with similar content (actually, in the definition of [13], there is no channel content). We refer here to the definition of [7], where compositional HMSCs generate weak-FIFO MSCs. Note that so far, compositional HMSCs do not allow for non-weak FIFO description, but that a small variant of the language could easily be defined to allow this kind of description (as long as non FIFOness remains inside a node of the CHMSC).

A CHMSC H generates a set of MSCs, denoted $Vis(H)$ by abuse of notation, obtained by concatenation of compositional MSCs along a path of the graph. With this definition, some path of a CHMSC may not generate any correct MSC. Moreover, a path of a CHMSC generates at most one MSC. The class of CHMSC for which each path generates exactly one MSC is called *safe* CHMSC, still a strict extension over HMSCs. S-regular and globally-cooperative HMSCs have also their strict extensions in terms of safe CHMSCs, namely as s-regular CHMSC and globally-cooperative CHMSCs.

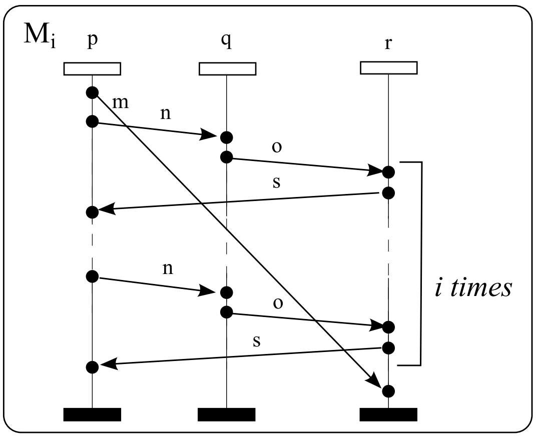

Fig. 15. A regular (but not finitely generated) set of MSCs

Let us now compare causal HMSCs and CHMSCs. It is not hard to build a regular compositional HMSC which MSC language can not be defined with a causal HMSC. An example is a CHMSC H that generates the visual language

$Vis(H) = \{M_i \mid i = 0, 1, \ldots\}$, where each M_i consists of an emission of a message m from p to r, then a sequence of i blocks of three messages: a message n from p to q followed by a message o from q to r then a message s from r to p. And at last the reception of message m on r. This MSC language is represented in Figure 15. This visual language can be easily defined with a CHMSC, by separating emission and reception of m and iterating a MSC containing messages n, o, s an arbitrary number of times. Clearly, this $Vis(H)$ is not finitely generated, and it is not either the visual language of a causal HMSC. Assume for contradiction, that there exists a causal HMSC G with $Vis(G) = Vis(H)$. Let k be the number of messages of the biggest causal MSC which labels a transition of G. We know that M_{k+1} is in $Vis(G)$, hence $M_{k+1} \in Vis(\circledcirc(\rho))$ for some accepting path ρ of G. Let $N_1, \ldots, N_\ell$ be causal MSCs along ρ, where $\ell \geq 2$ because of the size k. It also means that there exist $N'_1 \in Vis(N_1)$, ..., $N'_\ell \in Vis(N_\ell)$ such that $N'_1 \circ \cdots \circ N'_m \in Vis(G)$. Thus, $N_1 \circ \cdots \circ N_\ell = M_j$ for some j, a contradiction since M_j is a basic part (i.e. cannot be the concatenation of two MSCs). That is (regular) compositional HMSCs are not included into causal HMSCs. On the other hand, regular causal HMSCs have a regular set of linearizations (Theorem 2). Also by the results in [14], it is immediate that the class of visual languages of regular compositional HMSCs captures all the MSC languages that have a regular set of linearizations. Hence the class of regular causal HMSCs is included into the class of regular compositional HMSCs. Last, we already know with Figure 11 that globally-cooperative causal HMSCs are not necessarily existentially bounded, hence they are not included into safe compositional HMSC.

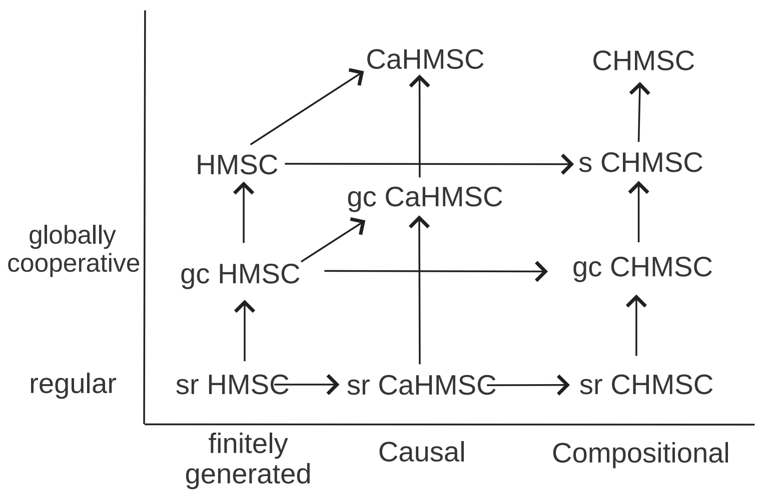

Fig. 16. Comparison of Scenario languages

The relationships among these scenario models are summarized by Figure 16, where arrows denote *strict* inclusion of visual languages. Two classes are incomparable if they are not connected by a transitive sequence of arrows. We use the abbreviation *sr* for s-regular, *gc* for globally-cooperative, *s* for safe, *CaHMSC* for causal HMSCs and *CHMSC* for compositional HMSCs.

6 A Detailed Example

We consider the TCP sliding window mechanism to show the usefulness and the expressive power of causal HMSCs.

The *Transmission Control Protocol* (TCP) is one of the core protocols of Internet. Using TCP, applications on networked hosts can create point-to-point connections to one another, over which they can exchange data in packets. The protocol guarantees reliable and in-order delivery of data from sender to receiver. TCP also distinguishes data for multiple connections by concurrent applications (e.g. Web server and e-mail server) running on the same host.

We first explain the relevant aspects of the TCP protocol [24]. For reasons of simplicity and readability, we shall abstract away some of the technical aspects of the protocol when constructing a model. The classical TCP is divided into 3 parts:

The first one is *connection establishment*. The procedure to establish connections uses a synchronize (*syn*) packet and involves an exchange of three messages. This exchange is called a three-way handshake [6]. Once a connection is established it can be used to carry data in both directions, that is, the connection is "full duplex". This connection phase can be modeled using MSCs, as shown in Figure 17. In this example, MSC *Connect*12 describes the case when process 1 initiates a connection, and MSC *Connect*21 the case when process 2 initiates the connection. When a process executes an event labeled by *start*, it is ready to begin the data transfer phase of TCP.

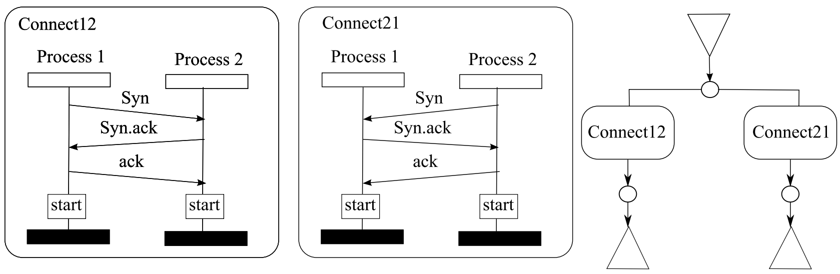

Fig. 17. Connection establishment between process 1 and process 2.

The second phase of the TCP protocol is *data transfer*. TCP is able to transfer a continuous stream of bytes in each direction between its users by packaging some number of bytes into segments for transmission through the Internet system. TCP uses sequence numbering in order to recover from data that is damaged, lost, duplicated, or delivered out of order by the Internet communication system. This is achieved by assigning a sequence number to each segment transmitted, and requiring a positive acknowledgment (ack) from the receiving tcp. Actually, the sequence number of ack sent by process p is the sequence number of the next tcp packet that p expects. Figure 18 shows a causal HMSC modeling a bi-directional data transfer, where sequence numbers are abstracted.

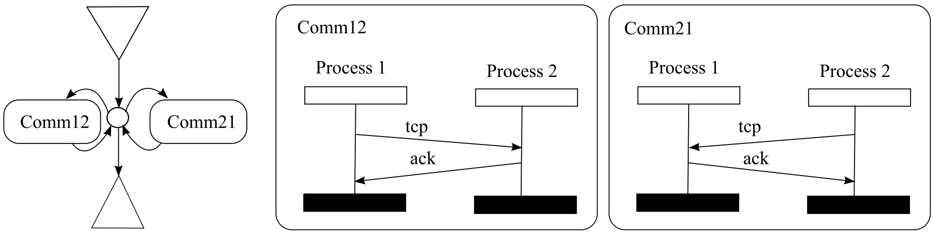

Fig. 18. Data transfer between process 1 and process 2.

The last phase of the TCP protocol is *connection termination*. The connection termination phase uses a four-way handshake. Each side of the connection terminates the session independently. When an endpoint wishes to stop its half of the connection, it transmits a fin packet, which the other end acknowledges with an ack. Therefore, a typical tear-down requires a pair of fin and ack segments from each tcp endpoint. The four-way handshake is modeled on figure 19: an end event is seen on process p when no more tcp packet are sent from p. In MSC $fin1$, process 1 stops first, and process 2 can continue to send tcp packets, then process 2 stops. In MSC $fin12$, process 1 and process 2 stops at the same time. In MSC $fin2$ process 2 initiates the termination of the communication, and then process 1 stops.

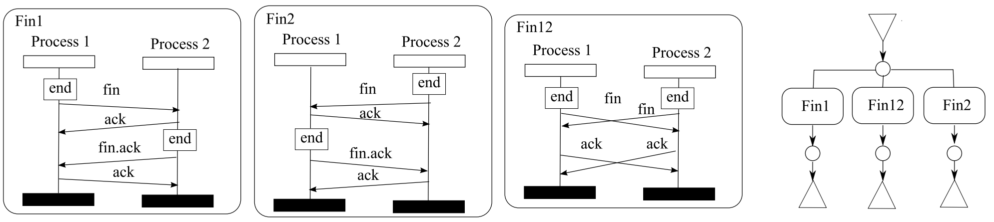

Fig. 19. The TCP connection termination.

A connection can be "half-open" when one side has terminated its connection, but not the other. The side that has terminated can no longer send data using this connection, but the other side can. Finally, it is possible for both hosts to send fin simultaneously. In this case, both sides just have to send ack packets to terminate the TCP connection. This can be considered as a 2-way handshake since the fin/ack sequence is done in parallel in both directions.

Automata of Figure 17, Figure 18 and Figure 19 model the 3 phases of TCP protocol. A complete description of the TCP protocol with MSCs can be obtained as a composition of these tree models, just by performing a classical sequential composition of the automata, as shown on Figure 20.

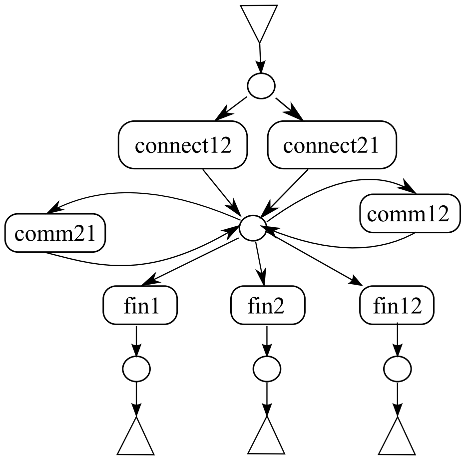

Fig. 20. A (causal) HMSC model of the TCP protocol.

So far, we have proposed scenario descriptions of the TCP protocol, and provided the HMSCs describing the typical executions of TCP, but we did not define the commutation relation over events of the protocol that allows for the interleaving of different phases of the protocol. Let us define the local dependency relations D_p for process p in $\{Process1, Process2\}$. If we chose the normal weak concatenation, as defined in HMSCs, i.e. $D_p = \Sigma_p \times \Sigma_p$, we obtain a synchronized execution of data transfer phase: every process send only one *tcp* packet and waits for the corresponding *ack* packet. Left part of figure 21 describes this kind of execution. This visual order is generated by the causal HMSC on figure 20 with $D_p = \Sigma_p \times \Sigma_p$ for each process, i.e. the usual weak concatenation of HMSC. Note however that in an implementation of the TCP protocol, data transfer from the two sites can be performed in parallel. Hence, the classical sequential composition of HMSCs does not suffice to model interesting behaviors of TCP. Moreover, processes can send *tcp* packets without waiting for acknowledgments. Thus, events occurring between $p(start)$ and $p(end)$ can occur in any order in a visual extension, i.e. $I_p = \{p!q(tcp), p?q(ack), p?q(tcp), p!q(ack), p?q(fin)\}^2 - \{(a, a) \in \Sigma_p^2\}$ for each process p in $\{Process1, Process2\}$ and q in $\{Process1, Process2\} \setminus \{p\}$. An execution of this causal HMSC is shown on the right part of figure 21.

Note that the causal HMSC we propose in Figure 20 for modeling the TCP protocol is not a regular causal HMSC and its linearization language is not regular: that means this protocol cannot be modeled by classical communicating finite state machines. Furthermore, this model is not window bounded, as an infinite number of *ack* messages can cross *tcp* ones. Finally, this causal HMSC H is not globally-cooperative, as the cycle labeled by $comm12.comm21$ does not have a connected communication graph. However, it is possible [9]

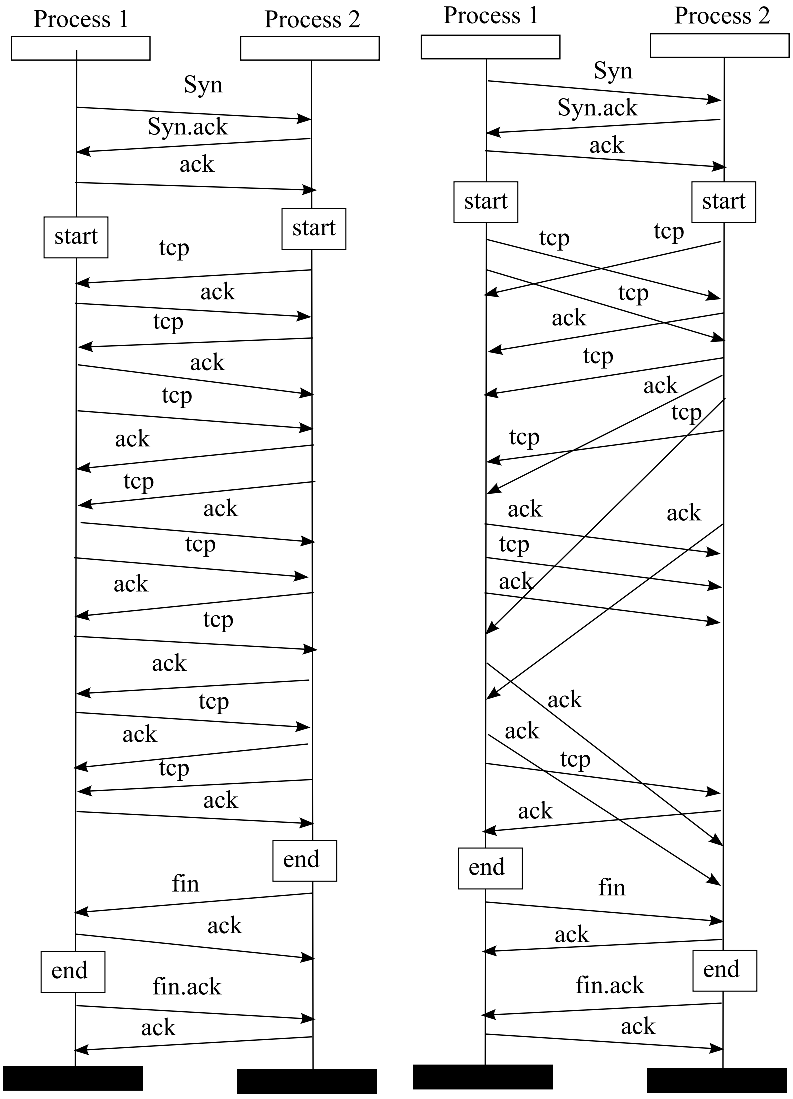

Fig. 21. Two executions of the TCP example, with the HMSC semantics (on the left) and with the causal HMSC semantics (on the right).

to show that the language of H viewed as an automaton on basic parts is closed by commutation, that is $\mathcal{L}_{Basic}(H) = [\mathcal{L}_{Basic}(H)]$. In this case, we can apply directly the second part of Theorem 3 with $\mathcal{A}'$ equals to H'. Consequently, this protocol can be model-checked against any property described by a causal HMSC.

7 Conclusion

We have defined an extension of HMSC called causal HMSC that allows the definition of braids, such as those appearing in sliding window protocols. We have also identified in this setting, many subclasses of scenarios that were defined for HMSCs which have decidable verification problems. An interesting class that emerges is globally-cooperative causal HMSCs. This class is incomparable with safe compositional HMSCs because the former can generate scenario collections that are not existentially bounded. Yet, decidability results of verification can be obtained for this class.

An interesting open problem is deciding whether the visual language of a causal HMSC is finitely generated. Yet another interesting issue is to consider the class of causal HMSCs whose visual languages are window-bounded. The set of behaviors generated by these causal HMSCs seems to exhibit a kind of regularity that could be exploited. Finally, designing suitable machine models (along the lines of Communicating Finite Automata [5]) is also an important future line of research.

References

[1] M. Ahuja, A.D. Kshemkalyani, and T. Carlson. A Basic Unit of Computation in Distributed Sytems. In *Proc. of ICDS'90*, pages 12–19, 1990.

[2] R. Alur, G.J. Holzmann, and D. Peled. An analyzer for Message Sequence Charts. In *Proc. of TACAS'96*, number 1055 in LNCS, pages 35–48, 1996.

[3] R. Alur and M. Yannakakis. Model checking of Message sequence Charts. In *Proc. of CONCUR'99*, number 1664 in LNCS, pages 114–129. Springer, 1999.

[4] B. Bollig. On the Expressiveness of Asynchronous Cellular Automata. In *FCT*, pages 528–539, 2005.

[5] D. Brand and P. Zafiropoulo. On Communicating Finite State Machines. Technical Report RZ1053, IBM Zurich Research Lab, 1981.

[6] Y. Dalal and C. Sunshine. Connection Management in Transport Protocols. *Computer Networks*, 2(6):454–473, 1978.

[7] P. Darondeau, B. Genest, and L. Hélouet. Products of message sequence charts. In *FOSSACS 2008*, number 4962 in LNCS, pages 459–474, 2008.

[8] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, 1995.

[9] T. Gazagnaire. *Scenario Languages: Using Partial Orders to Model, Verify and Supervize Distributed and Concurrent Systems*. PhD thesis, Université Rennes 1, 2008.

[10] T. Gazagnaire, B. Genest, L. Hélouët, P.S. Thiagarajan, and S. Yang. Causal Message Sequence Charts. In *CONCUR*, pages 166–180, 2007.

[11] B. Genest, D. Kuske, and A. Muscholl. A Kleene Theorem and Model Checking for a Class of Communicating Automata. *Information and Computation.*, 204(6):920–956, 2006.

[12] B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-state High-level MSCs: Model-checking and Realizability. *Journal of Computer and System Sciences*, 72(4):617–647, 2006.

[13] E. Gunter, A. Muscholl, and D. Peled. Compositional Message Sequence Charts. In *Proc. of TACAS'01*, number 2031 in LNCS. Springer, 2001.

[14] J.G. Henriksen, M. Mukund, K. Narayan Kumar, M. Sohoni, and P.S. Thiagarajan. A Theory of Regular MSC Languages. *Information and Computation*, 202(1):1–38, 2005.

[15] L. Hélouët and P. Le Maigat. Decomposition of Message Sequence Charts. In *Proc. of SAM'00*, 2000.

[16] ITU-TS. *ITU-TS Recommendation Z.120: Message Sequence Chart (MSC)*. ITU-TS, 1999.

[17] D.J Kleitman and B.L Rotschild. Asymptotic enumeration of partial orders on a finite set. *Transactions of the American Mathematical Society*, (205):205–220, 1975.

[18] D. Kuske. Regular sets of infinite message sequence charts. *Information and Computation*, 187(1):80–109, 2003.

[19] R. Morin. Recognizable sets of Message Sequence Charts. In *Proc. of STACS'02*, number 2285 in LNCS, pages 523–534. Springer, 2002.

[20] A. Muscholl and D. Peled. Message Sequence Graphs and Decision Problems on Mazurkiewicz Traces. In *Proc. of MFCS'99*, number 1672 in LNCS. Springer, 1999.

[21] A. Muscholl, D. Peled, and Z. Su. Deciding Properties for Message Sequence Charts. In *Proc. of FoSSaCS'98*, number 1378 in LNCS, pages 226–242. Springer, 1998.

[22] M. Reniers. *Message Sequence Chart: Syntax and Semantics*. PhD thesis, Eindhoven University of Technology, 1999.

[23] M. Reniers and S. Mauw. High-level Message Sequence Charts. In A. Cavalli and A. Sarma, editors, *SDL97: Time for Testing - SDL, MSC and Trends*, Proc. of the 8th SDL Forum, pages 291–306, Evry, France, Septembre 1997.

[24] RFC793. Transmission control protocol, 1981. DARPA Internet Program Protocol Specification.

[25] E. Rudolph, P. Graubman, and J. Grabowski. Tutorial On Message Sequence Charts. *Computer Networks and ISDN Systems*, 28:1629–1641, 1996.

[26] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2), 1972.